

FUJITSU Software

BS2000 OSD/BC V10.0

Commands

Volume 2: ACTIVATE-SNAPSHOT – DECRYPT-FILE

Valid for

SDF V4.7D

SDF-P-BASYS V2.5E

ASE V1.0B

BLSSERV V2.8A

CONV2PDF V1.0B

DSSM V4.3B

IMON-GPN V3.3A

JV V15.1A

POSIX-BC V10.0A

RFA V19.0A

RSO V3.6A

SECOS V5.4A

SPACEPRO V1.0A

SPOOL V4.9A

Comments... Suggestions... Corrections...

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:

manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH

www.cognitas.de

Copyright and Trademarks

Copyright © 2017 Fujitsu Technology Solutions_GmbH.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

ACTIVATE-SNAPSHOT

Start dump generator SNAP

Description status:	BS2000 OSD/BC V10.0A
Functional area:	System control and optimization
Domain:	SYSTEM-MANAGEMENT
Privileges:	TSOS OPERATING
Routing code:	R

Function

The ACTIVATE-SNAPSHOT enables the SNAP dump function. Any suspended SNAP dump will be processed (i.e. it will be assigned index structures for evaluation with the DAMP program and copied to the SYSSNAP ID). The command can be executed multiple times, e.g. in order to modify the size of the system file \$TSOS.SNAPFILE.

ACTIVATE-SNAPSHOT is executed asynchronously. Message NSP4000 confirms that the command has been received correctly. The SHOW-SNAPSHOT-STATUS command enables you to check the modified settings.



Manual changes to the Snapshot files can lead to undefined statuses when the SNAP dump takes place. Modifications may only be made using the ACTIVATE-SNAPSHOT and DEACTIVATE-SNAPSHOT commands.

Format

ACTIVATE-SNAPSHOT

FILE-SIZE = *UNCHANGED-OR-STD / <integer 16..1024 *Mbyte*>

Operands

FILE-SIZE = *UNCHANGED-OR-STD / <integer 16..1024> *Mbyte*

Determines the size of the system file \$TSOS.SNAPFILE on the home pubset.

FILE-SIZE = *UNCHANGED-OR-STD

The size of the system file \$TSOS.SNAPFILE remains unchanged. If it does not yet exist, it is created with the standard size of 144 MB.

FILE-SIZE = <integer 16..1024 Mbyte>

Size of the system file \$TSOS.SNAPFILE on the home pubset. The system file is expanded or reduced as required. If it does not yet exist, a new system file is created with this size.



A size of at least 144 MB is recommended.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without error

ADAPT-SNAPSET-ACCESS

Adapt access to Snapsets in the case of remote mirroring

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	SNAPSET MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS HSMS-ADMINISTRATION

Function

If access to the disks of the pubset is switched dynamically between the source and target controllers (by means of the AutoSwap function of HIPLEX-AF or explicitly using SHC-OSD commands), access to the Snapsets assigned to the pubset is not also switched automatically. The ADAPT-SNAPSET-ACCESS command ensures that the Snapsets assigned are still available after such a switchover without the pubset having to be exported.

When the command is called, a check is made to see whether access to the assigned Snapsets takes place in the same controller as access to the disks of the pubset. If this is not the case, the switchover is emulated for the Snapsets assigned to the pubset:

- The Snapsets currently attached are taken out of service.
- The Snapsets of the local controller are then attached.

To execute the command the pubset must be in the LOCAL-ACCESSIBLE status. In the case of shared pubsets, Snapset access is adapted for all sharers of the pubset.

Format

ADAPT-SNAPSET-ACCESS
PUBSET = <cat-id 1..4>

Operands

PUBSET = <cat-id 1..4>

Catalog ID of the pubset for which Snapset access is to be adapted.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed without error
	32	CMD0216	User does not have the necessary privilege
	64	DMS1386	Error in the memory request
	64	DMS138B	No MRSCAT entry
	64	DMS13D5	The specified Snapset does not exist on the pubset
1	64	DMS13D7	Internal error in Snapset management: Return code of GCF
4	64	DMS13D7	Internal error in Snapset management: Return code when setting/resetting the reconfiguration lock
6	64	DMS13D7	Internal error in Snapset management: Return code of SHC-OSD
7	64	DMS13D7	Internal error in Snapset management: Return code of CCOPY
	64	DMS13D8	The pubset has no Snapsets
	64	DMS13E5	Snapset catalog does not exist

ADD-ACS-SYSTEM-FILE

Declare ACS system file

Description status:	ACS V19.0A
Functional area:	File processing
Domain:	FILE
Privileges:	ACS-ADMINISTRATION

Function

The ADD-ACS-SYSTEM-FILE command is used by system administration to declare a new AC system file identifier, and to assign the file name and attributes to this identifier. Users who wish to load the entries from an AC system file into their virtual alias catalog can use this identifier, i.e. the symbolic name of the AC system file, to address the file. The SHOW-ACS-SYSTEM-FILES command enables the user to request that the list of available AC system files is displayed. However, the ACS administrator can specify, when setting the attributes, that output of the identifier and its associated file name is suppressed. The system administration can also specify that the identifier which is being declared is to act as the default AC system file. In this case, the entries in this file will be read if the user specifies the identifier as "STD" (in the LOAD-ALIAS-CATALOG command). If the specified identifier already exists for the AC system file, the existing definition will be replaced by the new one. The declarations relating to the AC system file remain in effect until shutdown. When the subsystem is unloaded, the declarations will only be deleted if the operand SUBSYSTEM-PARAMETER = 'RESET' is specified in the STOP-SUBSYSTEM command.

Format

ADD-ACS-SYSTEM-FILE

```

ALIAS-CATALOG-ID = <composed-name 1..20>
, FILE-NAME = <filename 1..54>
, ATTRIBUTES = *STD / list-poss(4): *SYSTEM-DEFAULT / *INVISIBLE / *SECRET-FILE-NAME /
*PRIVILEGED

```

Operands

ALIAS-CATALOG-ID = <composed-name 1..20>

The symbolic name under which the AC system file can be addressed by a LOAD-ALIAS-CATALOG command. The user can request display of a list of all the available AC system files by a SHOW-ACS-SYSTEM-FILES command.

FILE-NAME = <filename 1..54>

The fully qualified actual name of the AC system file, in which the entries are stored. Before the file is added into the list of AC system files, the FILE-NAME operand will, if necessary, be supplemented by the user ID and default catalog ID of the calling task.

ATTRIBUTES =

Declares the attributes to be given to the AC system file.

ATTRIBUTES = *STD

Preset value: The AC system file is not to be given any of the following attributes.

ATTRIBUTES = *SYSTEM-DEFAULT

The AC system file is to be used as the default AC system file. The entries from this file will then automatically be loaded into the user's alias catalog if the ALIAS-CATALOG-ID operand in the LOAD-ALIAS-CATALOG command has the value *STD. If more than one AC system file is given the attribute SYSTEM-DEFAULT, then the last assignment will always be the one which applies. If none of the AC system files is explicitly given this attribute, it is implicitly given to the first which was declared as a system file.

ATTRIBUTES = *INVISIBLE

Specifies that when the AC system files which are available to the nonprivileged user are output (SHOW-ACS-SYSTEM-FILES command), the entry for this file should not be displayed. The identifier of this AC system file can thus only be used by users to whom it has previously been notified, or who can invoke specially prepared procedures. If the SHOW-ACS-OPTIONS command is used to output the AC files which are loaded then, instead of its ID, the character '*' will be shown for this file if the caller is not the ACS administration.

ATTRIBUTES = *SECRET-FILE-NAME

Specifies that when the list of AC files which are available to nonprivileged users is output (SHOW-ACS-SYSTEM-FILES and SHOW-ACS-OPTIONS commands), instead of the file name of this AC system file the string '*SYSTEM' is to appear. The (nonprivileged) user thus has no way of revealing the actual name of the file.

ATTRIBUTES = *PRIVILEGED

Specifies that when the AC system file is used (LOAD-ALIAS-CATALOG ALIAS-CATALOG-ID = <id> command), its entries are to be copied into the virtual alias catalog of the task as **system entries**.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed normally
	32	CMD0221	Internal error
	64	ACS0013	Invalid ACSF declaration
	130	ACS0036	Resource bottleneck

ADD-ALIAS-CATALOG-ENTRY

Add entry to current alias catalog

Description status:	ACS V19.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING ACS-ADMINISTRATION SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

The ADD-ALIAS-CATALOG-ENTRY command adds an entry into the current alias catalog of a task. Any existing entry for the same alias in the catalog is overwritten.

If no alias catalog yet exists for the task, it is automatically set up with the first entry. The following attributes can be defined by the user for the new entry created:

- real file or job variable name to be used for the alias name
- whether the alias substitution is to be logged
- whether the entry is displayed in the output of the SHOW-ALIAS-CATALOG-ENTRY command
- whether the entry is protected against changes or deletion by the MODIFY-ALIAS-CATALOG-ENTRY or REMOVE-ALIAS-CATALOG-ENTRY command
- the range of substitution: the entry should apply only for files, only for job variables, or for both (RANGE operand)

Entries created with ADD-ALIAS-CATALOG-ENTRY are only temporarily present in the task-local alias catalog, which means they are discarded when the alias catalog is deleted (explicitly with the PURGE-ALIAS-CATALOG command or implicitly when the task is terminated). If desired, however, such entries can be saved in a file by using the STORE-ALIAS-CATALOG command.

Privileged functions

When an individual catalog entry is being created, only the ACS administrator can explicitly make a distinction between user and system entries (TYPE operand). An alias explicitly created as a system entry is subject to less rigorous checking: it is always permissible to specify a catalog ID and user ID for it (except \$SYS* and \$TSOS).

Format

```

ADD-ALIAS-CATALOG-ENTRY

ALIAS-FILE-NAME = <filename 1..54>
,FILE-NAME = <filename 1..54> / *SAME
,ATTRIBUTES = *STD / *PARAMETERS(...)
    *PARAMETERS(...)
        LOGGING = *NO / *YES
        ,VISIBILITY = *YES / *NO
        ,PROTECTION = *NO / *YES
,TYPE = *USER / *SYSTEM
,RANGE = *STD / *FILE / *JV / *BOTH

```

Operands

ALIAS-FILE-NAME = <filename 1..54>

Alias for which the actual file or job variable name is to be replaced by ACS.

Not permitted are names of temporary files or job variables, TSOS user ID files or job variables, files or job variables of a user ID beginning with *SYS* and file names containing a specification of a generation or version. The ACS administrator can impose a general inhibit on the specification of a catalog and/or user ID (see the MODIFY-ACS-OPTIONS command). Regardless of this, the user ID of the current task can always be contained in alias names.

FILE-NAME = <filename 1..54>

Actual file or job variable name to be substituted by ACS for the alias. The name of a temporary file is not allowed, nor is it permitted to specify a generation or version. If the actual name is identical to the alias, the entry is treated as a normal alias definition (in contrast to *SAME). The replacement of the file name is logged as ACS0000. The substitution of the job variable name is logged with ACS0049.

FILE-NAME = *SAME

The actual file or job variable name is the same as the alias. That is why no substitution takes place in this case. However, the entry prevents the insertion of the defined prefix (see SET-FILE-NAME-PREFIX command). The substitution of the file or job variable name is logged.

ATTRIBUTES = *STD / *PARAMETERS(...)

Defines the attributes of the new entry created in the alias catalog.

ATTRIBUTES = *STD

Specifies that the entry in the alias catalog is to have the following attributes:

- LOGGING=*NO
- VISIBILITY=*YES
- PROTECTION=*NO

ATTRIBUTES = *PARAMETERS(...)

Attributes of the new entry are to be defined.

LOGGING = *NO / *YES

Logs substitution of the name alias. This attribute is independent of the corresponding ACS setting which applies to the task.

VISIBILITY = *YES / *NO

Prevents the entry in the alias catalog from appearing in the output of the SHOW-ALIAS-CATALOG-ENTRY command.

PROTECTION = *NO / *YES

Protects the entry in the alias catalog against changes and deletion with the MODIFY-ALIAS-CATALOG-ENTRY and REMOVE-ALIAS-CATALOG-ENTRY commands respectively.

TYPE = *USER / *SYSTEM

This operand is available only to users with the ACS-ADMINISTRATION privilege.

Indicates whether the specified entry is to be of the type USER or SYSTEM.

TYPE = *USER

Preset value: The entry made in the catalog will be a user entry. When the virtual catalog is saved, this entry will be included in the file records.

TYPE = *SYSTEM

The catalog entry is to be made as a system entry. If the catalog is saved to a nonprivileged user's file, this entry will not be included.

RANGE = *STD / *FILE / *JV / *BOTH

Determines the range of the entry. Alias substitution can be defined for files and/or job variables.

RANGE = *STD

Default: Alias substitution takes place using the ACS setting which applies for the task (ACS option STANDARD-RANGE).

RANGE = *FILE

Alias substitution is only effective for files.

RANGE = *JV

Alias substitution is only effective for job variables.

RANGE = *BOTH

Alias substitution is effective for both files and job variables.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed normally
	64	ACS0024	Invalid alias catalog entry
	64	ACS0029	Command is not permitted
	128	ACS0018	ACS is not available
	130	ACS0036	Resource bottleneck

Examples

Example 1: Load alias catalog, change entries, substitution function

```

/show-acs-system-file _____ (1)
% ALIAS CATALOG ID FILE NAME
%P >ACS-FOR-USER1 $TSOS.OTTY.XY

/show-alias-cat _____ (2)
% ACS0017 ALIAS CATALOG NOT AVAILABLE. COMMAND REJECTED

/load-alias-cat alias-cat=acs-for-user1 _____ (3)
% ACS0001 ALIAS CATALOG 'ACS-FOR-USER1' LOADED (DESCRIPTOR='ACS0193', DATE=
2012-02-09, NUMBER OF ENTRIES=2)

/show-alias-cat _____ (4)
% ALIAS FILE NAME -> FILE NAME
%SB : COB85 -> $RZ.COBOL85
%SB : FORTRAN -> $RZ.FOR1
% ACS0037 NUMBER OF ALIAS CATALOG ENTRIES: 2 (FOR SYSTEM: 2, FOR USER: 0)

/show-file-attr fortran _____ (5)
% 1266 :2ORZ:$RZ.FOR1
%:2ORZ: PUBLIC: 1 FILE RES= 1266 FRE= 0 REL= 0 PAGES

/add-alias alias=ass,file=$.assembh _____ (6)

/show-alias-cat _____ (7)
% ALIAS FILE NAME -> FILE NAME
%UB : ASS -> $.ASSEMBH
%SB : COB85 -> $RZ.COBOL85
%SB : FORTRAN -> $RZ.FOR1
% ACS0037 NUMBER OF ALIAS CATALOG ENTRIES: 2 (FOR SYSTEM: 2, FOR USER: 1)

/add-alias-cat alias=sf.sdf,file=$TSOS.SYS.SDF.SYSTEM.SYNTAX _____ (8)
/show-alias-cat _____ (9)
% ALIAS FILE NAME -> FILE NAME
%UB : ASS -> $.ASSEMBH
%SB : COB85 -> $RZ.COBOL85
%SB : FORTRAN -> $RZ.FOR1
%UB : SF.SDF -> $TSOS.SYS.SDF.SYSTEM.SYNTAX
% ACS0037 NUMBER OF ALIAS CATALOG ENTRIES: 4 (FOR SYSTEM: 2, FOR USER: 2)

/add-alias-cat alias=bsp,file=lst.bsp.2 _____ (10)

/show-alias-cat _____ (11)
% ALIAS FILE NAME -> FILE NAME
%UB : ASS -> $.ASSEMBH
%UB : BSP -> LST.BSP.2
%SB : COB85 -> $RZ.COBOL85
%SB : FORTRAN -> $RZ.FOR1
%UB : SF.SDF -> $TSOS.SYS.SDF.SYSTEM.SYNTAX
% ACS0037 NUMBER OF ALIAS CATALOG ENTRIES: 5 (FOR SYSTEM: 2, FOR USER: 3)

/show-file-attr sf.sdf _____ (12)
% 5328 :2OST:$TSOS.SYS.SDF.SYSTEM.SYNTAX
%:2OST: PUBLIC: 1 FILE RES= 5328 FRE= 1 REL= 0 PAGES

/show-file-attr sf. _____ (13)
% 384 :20SG:$USER1.SF.NEU
% 48 :20SG:$USER1.SF.ROBAR
% 123 :20SG:$USER1.SF.TEST.DEV.1
%:20SG: PUBLIC: 3 FILES RES= 555 FRE= 24 REL= 3 PAGES

```

```

/show-acs-opt _____ (14)
% ALIAS CATALOG SYSTEM V18.0
% =====
%
% STATUS: ACTIVE
%
% ACTIVATED ALIAS CATALOG FILE(S):
% ID          U-INFO  DESCR   DATE      FILE NAME
% ACS-FOR-USER1  SYSADM04 ACS0193 2012-02-09 :20ST:$TSOS.OTTY.XY
% ENTRIES ADDED/MODIFIED BY INDIVIDUAL COMMANDS:      3
%
% LOGGING: ALIAS-SUBSTITUTION=STD, PREFIX-INSERTION=NO
% SUCCESS-MSG OPTIONS: USER-FILE=YES, SYSTEM-FILE=YES
% COMPLETE-ALIAS-NAMES=NOT-ALLOWED (USER-MODIF=NOT-ALLOWED)
% ALIAS-USERID   =ALLOWED      (USER-MODIF=ALLOWED      )
% STANDARD-RANGE=BOTH
/print-document bsp _____ (15)
% SCP0860 FILE ':20SG:$USER1.LST.BSP.2' PROTECTED BY A READ OR EXEC PASSWORD.
COMMAND REJECTED

/load-exe from-file=cob85 _____ (16)
% COB9000 COPYRIGHT (C) ...
                        ALL RIGHTS RESERVED

/show-job-status _____ (17)
%TSN: 6XJE      TYPE: 3 DIALOG  NOW: 2012-02-09.183437
%JOBNAME: ALF    PRI: 0 210
%USERID: USER1  JCLASS: JCDSTD  LOGON: 2012-02-09.1758
%ACCNB: 89001   CPU-MAX: 9999   CPU-USED:000000.7431
%STATION: $$06580 PROC: FIREBALL
%TID: 000801FD UNP/Q#: 00/000
%CMD: SHOW-JOB-STATUS SIZE: 1971
%PROG: :20RZ:$RZ.COBOL85
%MONJV: *NONE

```

- (1) The SHOW-ACS-SYSTEM-FILE command shows the IDs of the available AC system files.
- (2) The SHOW-ALIAS-CATALOG-ENTRY command shows that no task-local alias catalog is loaded.
- (3) The LOAD-ALIAS-CATALOG command loads system entries from the *ACS-FOR-USER1* AC system file and activates the ACS replacement function.
- (4) The SHOW-ALIAS-CATALOG-ENTRY command shows that two alias name definitions have been loaded.
- (5) In the SHOW-FILE-ATTRIBUTES command the alias name *FORTRAN* is replaced by the defined real file name *\$RZ.FORI* before the command is executed.
- (6) The ADD-ALIAS-CATALOG-ENTRY command adds to the alias catalog a definition containing the alias name *ASS* for the file *\$.ASSEMBH* (user entry).
- (7) Shows the entries in the alias catalogs.
- (8) The ADD-ALIAS-CATALOG-ENTRY command adds to the alias catalog a definition containing the alias name *SF.SDF* for the file *\$TSOS.SYS.SDF.SYSTEM.SYNTAX* (user entry).

- (9) Shows the entries in the alias catalogs.
- (10) The ADD-ALIAS-CATALOG-ENTRY command adds to the alias catalog a definition with the alias name *BSP* for the file *LST.BSP.2* (user entry).
- (11) Shows the entries in the alias catalogs.
- (12) In the SHOW-FILE-ATTRIBUTES command the alias name *SF.SYS* is replaced by the defined real file name *\$TSOS.SDF.SYS.SYSTEM.SYNTAX* before the command is executed.
- (13) As the specified name *SF* does not correspond to an alias name, the SHOW-FILE-ATTRIBUTES command is used to display all files whose real file name begins with *SF*.
- (14) Shows the current ACS settings with the SHOW-ACS-OPTIONS command.
- (15) In the PRINT-DOCUMENT command the alias name *BSP* is replaced by the defined real file name *LST.BSP.2* before the command is executed.
- (16) The LOAD-EXECUTABLE-PROGRAM command loads a program from the file *COB85*. The message *BLS0500* shows that the program *COBOL85* has been loaded.
- (17) The SHOW-JOB-STATUS command shows that the program *COBOL85* has been loaded from the file *:2ORZ:\$RZ.COBOL*. The name *COB85* specified in LOAD-EXECUTABLE-PROGRAM corresponds to a defined alias name. It was therefore replaced by the assigned real file name *\$RZ.COBOL85* before the command was executed.

Example 2: Save alias catalog, change entries

```

/show-alias-cat _____ (1)
% ALIAS FILE NAME -> FILE NAME
%UB : ASS -> $.ASSEMBH
%UB : BSP -> LST.BSP.2
%SB : COB85 -> $RZ.COBOL85
%SB : FORTRAN -> $RZ.FOR1
%UB : SF.SDF -> $TSOS.SYS.SDF.SYSTEM.SYNTAX
% ACS0037 NUMBER OF ALIAS CATALOG ENTRIES: 5 (FOR SYSTEM: 2, FOR USER: 3)
/store-alias-cat to-file=acs.al,user-inf=userlac,descript=v0001a _____ (2)
% ACS0038 NUMBER OF ALIAS CATALOG ENTRIES STORED IN FILE ':20SG:$USER1.ACS.
A1': 3

/rem-alias-cat alias=cob85 _____ (3)
/rem-alias-cat alias=ass _____ (4)
/show-alias _____ (5)
% ALIAS FILE NAME -> FILE NAME
%UB : BSP -> LST.BSP.2
%SB : FORTRAN -> $RZ.FOR1
%UB : SF.SDF -> $TSOS.SYS.SDF.SYSTEM.SYNTAX
% ACS0037 NUMBER OF ALIAS CATALOG ENTRIES: 3 (FOR SYSTEM: 1, FOR USER: 2)

/mod-alias-cat alias=sf.sdf,attr=*invisible _____ (6)
/show-alias _____ (7)
% ALIAS FILE NAME -> FILE NAME
%UB : BSP -> LST.BSP.2
%SB : FORTRAN -> $RZ.FOR1
% ACS0037 NUMBER OF ALIAS CATALOG ENTRIES: 3 (FOR SYSTEM: 1, FOR USER: 2)

/mod-alias-cat alias=sf.sdf,attr=*protect _____ (8)
/show-alias-cat _____ (9)
% ALIAS FILE NAME -> FILE NAME
%UB : BSP -> LST.BSP.2
%SB : FORTRAN -> $RZ.FOR1
%UBP: SF.SDF -> $TSOS.SYS.SDF.SYSTEM.SYNTAX
% ACS0037 NUMBER OF ALIAS CATALOG ENTRIES: 3 (FOR SYSTEM: 1, FOR USER: 2)

/rem-alias-cat alias=sf.sdf _____ (10)
% ACS0005 ALIAS CATALOG ENTRY CANNOT BE MODIFIED. COMMAND REJECTED

/mod-alias-cat alias=sf.sdf,attr=*std _____ (11)
% ACS0005 ALIAS CATALOG ENTRY CANNOT BE MODIFIED. COMMAND REJECTED

/mod-alias-cat alias=bsp,file=max.file.3 _____ (12)

/add-alias-cat alias=max,file=max.file.3 _____ (13)

/show-alias select=*user _____ (14)
% ALIAS FILE NAME -> FILE NAME
%UB : BSP -> MAX.FILE.3
%UB : MAX -> MAX.FILE.3
%UBP: SF.SDF -> $TSOS.SYS.SDF.SYSTEM.SYNTAX
% ACS0037 NUMBER OF ALIAS CATALOG ENTRIES: 3 (FOR SYSTEM: 0, FOR USER: 3)

/load-alias-cat alias-cat=*own(acs.al),mode=*replace _____ (15)
% ACS0001 ALIAS CATALOG 'USER1AC' LOADED (DESCRIPTOR='V0001A', DATE=2011-02
-11, NUMBER OF ENTRIES=3)

/show-alias _____ (16)
% ALIAS FILE NAME -> FILE NAME
%UB : ASS -> $.ASSEMBH
%UB : BSP -> LST.BSP.2
%UB : SF.SDF -> $TSOS.SYS.SDF.SYSTEM.SYNTAX
% ACS0037 NUMBER OF ALIAS CATALOG ENTRIES: 3 (FOR SYSTEM: 0, FOR USER: 3)

```

- (1) The SHOW-ALIAS-CATALOG-ENTRY command shows all entries in the task-local alias catalog.
- (2) The STORE-ALIAS-CATALOG command saves all user entries in the alias catalog into the AC file *ACS.AI*.
- (3) Delete the entry with the alias name *COB85*.
- (4) Delete the entry with the alias name *ASS*.
- (5) The SHOW-ALIAS-CATALOG-ENTRY command displays all entries in the task-local alias catalog. There remain only the three entries with the alias names *BSP*, *FORTRAN* and *SF.SDF*.
- (6) For the entry with the alias name *SF.SDF* the attribute INVISIBLE in the MODIFY-ALIAS-CATALOG-ENTRY command is defined.
- (7) On output with SHOW-ALIAS-CATALOG-ENTRY the entry with the alias name *SF.SDF* is not shown. The total line is however shown for all existing entries.
- (8) For the entry with the alias name *SF.SDF* the attribute PROTECTED is defined in the MODIFY-ALIAS-CATALOG-ENTRY command. The attribute INVISIBLE is canceled. The attribute INVISIBLE would only be retained if explicitly specified, e.g. ATTRIBUTES=(*PROTECTED , *INVISIBLE).
- (9) In the output with the SHOW-ALIAS-CATALOG-ENTRY command the entry with the alias name *SF.SDF* is flagged with the letter *P* (for PROTECTED).
- (10) The entry with the alias name *SF.SDF* is to be deleted. The REMOVE-ALIAS-CATALOG-ENTRY command is rejected due to the attribute PROTECTED.
- (11) The entry with the alias name *SF.SDF* is to be changed. The MODIFY-ALIAS-CATALOG-ENTRY command is rejected due to the attribute PROTECTED.
- (12) For the entry *BSP* a new real file name (*MAX.FILE.3*) is defined with the MODIFY-ALIAS-CATALOG-ENTRY command.
- (13) With the ADD-ALIAS-CATALOG-ENTRY command a new entry with the alias name *MAX* is defined for the file *MAX.FILE.3*. The file *MAX.FILE.3* can now be accessed via both *MAX* and *BSP*.
- (14) Shows the user entries in the alias catalog.
- (15) The entries in the AC file *ACS.AI* are loaded into the alias catalog. All previous entries are thereby removed (MODE=*REPLACE).
- (16) Output with the SHOW-ALIAS-CATALOG-ENTRY command shows only the three user IDs saved with STORE-ALIAS-CATALOG-ENTRY to the AC file *ACS.AI* (see 1 and 2 above). The system entries and all changes in the alias catalog made after saving have been removed.

For further examples, see the HOLD-ALIAS-SUBSTITUTION, LOAD-ALIAS-CATALOG and SET-FILE-NAME-PREFIX commands.

ADD-ASE-ELEMENT

Declare an ASE element

Description status:	ASE V1.0B
Functional area:	Error logging
Domain:	ERROR-LOGGING
Privileges:	TSOS

Function

The ADD-ASE-ELEMENT command defines, for a particular set of SERSLOG events, monitoring which triggers specific actions when a defined threshold value for the occurrence of these events is reached. The set of events (RECORD-ID operand) can be defined as a

- selection of all messages, with the option of excluding message classes or messages,
- selection of one or more specified messages,
- selection of a message class, with the option of specifying limitation to one message range and/or of excluding messages.

The defined set of SERSLOG events forms an ASE element within the ASE (Auxiliary SERSLOG Extensions) subsystem for which particular criteria for monitoring and triggering actions are defined in the declaration:

- For monitoring purposes, a threshold value (THRESHOLD operand) defines the number of events which can occur before a renewed occurrence will trigger actions. It is also possible to define whether and how the repeated occurrence of the same events is to be handled (SUPPRESS-DUPLICATES, AFTER and SLEEP-TIME operands).
- The actions to be triggered (ACTION operand) which can be defined are the logging of the event in the internal buffer, output of a console message or triggering of a Teleservice call. Multiple actions can be combined.
- Monitoring can take place for all tasks or be limited to one particular task (TASK operand).

After the declaration has been made, ASE assigns a unique element ID for the ASE element. This ID enables the ASE element to be specified for deletion or for information output.

Information on ASE elements can be requested using the SHOW-ASE-ELEMENT command.

Format

ADD-ASE-ELEMENT
<pre> RECORD-ID = *ALL(...) / *CLASS(...) / list-poss(10): <alphanum-name 7..7> *ALL(...) EXCLUDE-CLASS = *NONE / list-poss(10): <alphanum-name 3..3> EXCLUDE-RECORD = *NONE / list-poss(10): <alphanum-name 7..7> *CLASS(...) RECORD-CLASS = <alphanum-name 3..3> FROM = *FIRST / <alphanum-name 4..4> TO = *LAST / <alphanum-name 4..4> EXCLUDE-RECORD = *NONE / list-poss(10): <alphanum-name 4..4> / <alphanum-name 7..7> ,THRESHOLD = *NONE / <integer 2..255> ,SUPPRESS-DUPLICATES = *NO / *YES(...) *YES(...) AFTER = 1 / <integer 1..9> SLEEP-TIME = *UNLIMITED / <integer 1..999 hours> / <time> ,ACTION = list-poss(3): *INTERNAL-LOGGING / *CONSOLE-MSG / *TELESERVICE-CALL ,TASK = *ALL / <alphanum-name 4..4> / <c-string 4..4> </pre>

Operands

RECORD-ID = *ALL(...) / ***CLASS(...)** / list-poss(10): <alphanum-name 7..7>

Determines the set of SERSLOG events which is to be monitored.

RECORD-ID = *ALL(...)

All SERSLOG events are to be monitored. Entire message classes or individual messages can then be excluded.

EXCLUDE-CLASS = *NONE / list-poss(10): <alphanum-name 3..3>

Specifies message classes which are to be excluded from monitoring. The default *NONE means that no message classes are excluded.

EXCLUDE-RECORD = *NONE / list-poss(10): <alphanum-name 7..7>

Specifies individual messages which are to be excluded from monitoring. The default *NONE means that no messages are excluded.

RECORD-ID = *CLASS(...)

One or all message ranges of a particular message class are to be monitored.

RECORD-CLASS = <alphanum-name 3..3>

Message class which is to be monitored.

FROM = *FIRST / <alphanum-name 4..4>

Specifies the lower limit of the range. The default *FIRST means that the range begins with the first message of the specified message class.

TO = *LAST / <alphanum-name 4..4>

Specifies the upper limit of the range. The default *LAST means that the range ends with the last message of the specified message class.

EXCLUDE-RECORD = *NONE / list-poss(10): <alphanum-name 4..4> / <alphanum-name 7..7>

Specifies individual messages of the specified range which are to be excluded from monitoring. The specification <alphanum-name 4..4> contains the message code which ends with this character string. The default *NONE means that no messages of the range are excluded.

THRESHOLD = *NONE / <integer 2..255>

Specifies a threshold value which, when it is reached, causes the actions specified in the ACTION operand to be executed.

THRESHOLD = *NONE

The first occurrence of a SERSLOG event from the set of record types specified in the RECORD-ID operand triggers actions.

THRESHOLD = <integer 2..255>

The specified number of SERSLOG events from the set specified in the RECORD-ID operand must have occurred in order to trigger actions.

SUPPRESS-DUPLICATES = *NO / *YES(...)

Defines how the repetition of SERSLOG events after the threshold value specified in the THRESHOLD operand has been reached is to be handled.

SUPPRESS-DUPLICATES = *NO

Repeated events trigger actions.

SUPPRESS-DUPLICATES = *YES(...)

Repeated events only trigger actions conditionally.

AFTER = 1 / <integer 1..9>

Determines the number of repetitions as of which the actions are to be suppressed. The default value 1 means that actions are suppressed from the first repetition on.

SLEEP-TIME = *UNLIMITED / <integer 1..999 hours> / <time>

Specifies how long actions are to be suppressed. After the specified period has elapsed, the count associated with the AFTER operand is reinitialized.

ACTION = list-poss(3): *INTERNAL-LOGGING / *CONSOLE-MSG / *TELESERVICE-CALL

Determines the actions which are to be executed when the specified trigger criteria are met. Up to three actions can be specified in a list.

ACTION = *INTERNAL-LOGGING

The event is logged in an internal buffer whose content can be displayed using the SHOW-ASE-LOGGING command.

ACTION = *CONSOLE-MSG

Console message ASE0815 is output.

ACTION = *TELESERVICE-CALL

A Teleservice call is triggered (message ASE0888).

TASK = *ALL / <alphanum-name 4..4> / <c-string 4..4>

Specifies the TSN of the task which is to be monitored. The default *ALL means that all tasks are monitored.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed without error
	1	ASE0010	Syntax error
	32	CMD0221	System error
	64	ASE0011	Semantic error

ADD-CATALOG-FILE

Extend special catalog by another catalog file

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MANAGEMENT
Privileges:	TSOS

Function

The ADD-CATALOG-FILE command enables systems support to extend the special catalog of an SM pubset if the SM pubset has the new catalog format “extra large”.

When the command executes, another catalog file is created for the special catalog selected. The file name of the new catalog file has the following format:

```
:<catid>:$TSOS.TSOSCAT#xnn
```

where:

x	designates the special catalog (M, J or P; see CATALOG operand)
nn	consecutive numbering of the special catalog (00 through 99)

The numbering of the special catalogs never has any gaps. No file with this name may already exist.

The command can only be issued on the master processor.

Format

ADD-CATALOG-FILE
PUBSET-ID = <cat-id 1..4> ,CATALOG = *MIGRATED / *JV / *PRIVATE

Operands

PUBSET-ID = <cat-id 1..4>

Designates the pubset for which a special catalog is to be extended.

CATALOG = *MIGRATED / *JV / *PRIVATE

Defines which special catalog is to be extended.

CATALOG = *MIGRATED

The catalog for the migrated files and the no space files is extended, i.e. a file with the format :<catid>:\$TSOS.TSOSCAT.#Mnn is created.

CATALOG = *JV

The catalog for the job variables is extended, i.e. a file with the format :<catid>:\$TSOS.TSOSCAT.#Jnn is created.

CATALOG = *PRIVATE

The catalog for the private disk and tape files is extend, i.e. a file with the format :<catid>:\$TSOS.TSOSCAT.#Pnn is created.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed without error
	32	CMD0221	Internal error
	64	DMS0312	Catalog not known in system
	64	DMS0501	Catalog is not available in the system
	64	DMS134C	MRSCAT cannot be determined
	64	DMS13BB	Pubset is not master or not an SM pubset
	64	DMS13BC	File of hte same name already exists
	64	DMS13BE	Catalog is not known in the catalog index
	128	DMS0506	Change of master running

ADD-CHANGE-DATE

Define a new changeover time

Description status:	BS2000 OSD/BC V10.0A
Functional area:	System control and optimization
Domain:	SYSTEM-MANAGEMENT
Privileges:	TSOS

Function

The ADD-CHANGE-DATE command defines a new future changeover time from standard daylight time to daylight saving time (and vice versa).

A valid GTIME parameter set with consistent changeover times for the past is required, see the "Introduction to System Administration" [14].

The new changeover time must satisfy the following conditions:

- It must be later than all changeover times which have already been defined
- It may not be within the next hour
- It must be at least 4 to 8 months later than the previous changeover time



The ADD-CHANGE-DATE command defines the new changeover time for the current session. If required, also enter the new changeover time in the GTIME parameter set of the startup parameter file so that it is valid for future sessions.

Format

ADD-CHANGE-DATE
DATE = <date> ,TIME = <time>

Operands**DATE = <date>**

Date of the new changeover time.

TIME = <time>

Time of day of the new changeover time in the format hh:mm.

No seconds may be specified.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
	1	CHD0010	Syntax error in the command
	32	CMD0221	System error in the command
	64	CHD0011	Semantic error in the command

ADD-CJC-ACTION

Declare conditional execution of command sequence

Description status:	JV V15.1A
Functional area:	Job variables
Domain:	JOB-VARIABLES
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

This function is available to the user only if the chargeable software product JV has been loaded as a subsystem.

Function

Via the ADD-CJC-ACTION command, the user can make the execution of a CJC (conditional job control) command sequence dependent on a condition. The ADD-CJC-ACTION command defines

- the condition under which the CJC command sequence is to be started (operand CONDITION)
- the name of the CJC command sequence (operand NAME)
- the time period during which the command is to be in effect (operand TIME-LIMIT)
- the maximum number of executions of the CJC command sequence (operand REPEAT)
- the address (branch destination) of the CJC command sequence to be started in the case: time elapsed and “condition not satisfied” (operand TIMEOUT-LABEL)

The user enters the CJC command sequence following the ADD-CJC-ACTION command. The sequence may contain only ENTER-JOB, ENTER-PROCEDURE- and/or MODIFY-JV commands. ENTER-PROCEDURE is only permitted in a non-S procedure. The whole CJC command sequence between ADD-CJC-ACTION and END-CJC-ACTION is initially only stored. The following table illustrates the order in which the commands in the sequence are to be entered:

Input	Meaning
/ADD-CJC-ACTION_...	Opens a CJC command sequence; defines the condition and type of execution
$\left. \begin{array}{l} \text{ENTER-JOB} \\ \text{ENTER-PROCEDURE} \\ \text{MODIFY-JV} \\ \vdots \end{array} \right\} \dots$	CJC command sequence to be started in the case “condition satisfied”; merely stored initially. ENTER-PROCEDURE is only permitted in a non-S procedure.
$\left. \begin{array}{l} \text{.marke-} \\ \text{ENTER-JOB} \\ \text{ENTER-PROCEDURE} \\ \text{MODIFY-JV} \\ \vdots \end{array} \right\} \dots$	CJC (TIMEOUT) command sequence to be started in the case “condition not satisfied”; merely stored initially. This command sequence can also be defined. If it is used, the name of the command sequence (.marke) must be the same as the branch destination defined in ADD-CJC-ACTION (TIMEOUT-LABEL=marke). ENTER-PROCEDURE is only permitted in a non-S procedure. If the command sequence begins with an ENTER-JOB or ENTER-PROCEDURE command, a job name specified in the command must be the same as the name of the command sequence (JOB-NAME=marke).
/END-CJC-ACTION	Closes and activates the CJC command sequence

Table 18: Order of commands to be entered in the CJC command sequence

Both command sequences (“condition satisfied” or “condition not satisfied”) may be specified.

The END-CJC-ACTION command concludes the CJC command sequence and activates it. Subsequent commands are again executed immediately.

If “condition satisfied” occurs within the defined time period, processing is interrupted and the corresponding CJC command sequence is started. This applies for every “condition satisfied” until the specified time has elapsed or the maximum number of repetitions of the CJC command sequence has been reached. If the condition is already satisfied when the CJC command sequence is specified, the CJC commands are executed immediately.

If the time elapses without the maximum number of “condition satisfied” events having occurred, the TIMEOUT command sequence, if one has been defined, is started. When execution of a CJC command sequence has ended, processing continues from the point at which it was interrupted. If the CJC command sequence corresponding to the command ADD-CJC-ACTION is not given, only the event that fulfils the specified condition is logged by messages.

Format

ADD-CJC-ACTION

```
CONDITION = *NONE / <text 0..1800 with-low cond-expr>  
,NAME = *NONE / <name 1..8>  
,TIME-LIMIT = 600 / <integer 1..65535 seconds>  
,TIMEOUT-LABEL = *NONE / <name 1..8>  
,REPEAT = 1 / <integer 1..32767>
```

Operands

CONDITION = *NONE / <text 0..1800 with-low *cond-expr*>

Conditional expression (enclosed in parentheses). The CJC command sequence is executed as soon as “condition satisfied” occurs. The permissible entries are described in [section “Conditional expressions \(job variables\)” on page 1-70](#). Uppercase and lowercase letters are differentiated. Special job variables are not permitted here. The default setting is *NONE, i.e. only a TIMEOUT command sequence can be started (see the TIMEOUT-LABEL operand). Commands before the TIMEOUT mark and a REPEAT value greater than 1 are rejected.

NAME = *NONE / <name 1..8>

Name for the ADD-CJC-ACTION command. This name is displayed in messages beside the internal identification. The ADD-CJC-ACTION command can be referred to using this name. If the name specified is identical to the name of an ADD-CJC-ACTION command still in effect, the entry is accepted but a warning message is issued. The default is *NONE, i.e. messages show only the identification assigned by the system. The internal identification is then the only means of referring to the ADD-CJC-ACTION command.

TIME-LIMIT = 600 / <integer 1..65535 *seconds*>

Time specification, in seconds. The CJC command sequence is executed only if “condition satisfied” occurs within the specified time period and the maximum number of repetitions (REPEAT operand) is not exceeded.

The countdown begins after the END-CJC-ACTION command is executed; the average precision is 200 ms. A time specification greater than 65280 seconds means “unlimited life” (but not beyond job termination).

TIMEOUT-LABEL = *NONE / <name 1..8>

Branch destination. Name of the command with which processing is to be continued if the specified time runs out and the maximum number of “condition satisfied” events has not yet occurred (TIMEOUT command sequence).

The default value is *NONE, i.e. no TIMEOUT command sequence is specified.

If no TIMEOUT command sequence with the specified name is defined at the time END-CJC-ACTION is entered, then

- with procedures (interactive/batch): the whole CJC command sequence is rejected.
- in dialog (without procedure): a warning is issued and END-CJC-ACTION is rejected.

REPEAT = 1 / <integer 1..32767>

Maximum number of repetitions of the CJC command sequence within the specified period of time.

Return codes

Return codes relating to the ADD-CJC-ACTION command or the CJC command sequence commands are only returned once the CJC command sequence is terminated with END-CJC-ACTION.

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	CJC command sequence executed
	1	CMD0202	Condition incorrect, syntax error, CJC command sequence command not allowed or unknown, END-CJC-ACTION not found or timeout label does not exist
	32	CMD0221	Internal error
	130	CJC0002	JV not accessible
	130	CJC0003	Memory saturation

Notes

- A CJC command sequence is always viewed by the system as a self-contained unit, i.e. processing must not branch to a CJC command sequence from an external source. For every branching operation to be performed (to a mark or to SET-JOB-STEP, for instance), a search is made for an END-CJC-ACTION following any ADD-CJC-ACTION detected; only then will the search for the desired branch destination be continued.
- An ADD-CJC-ACTION command becomes inoperative in the following cases:
 - The CJC command sequence has been carried out as often as was specified in the REPEAT operand.
 - The time interval specified in the TIME-LIMIT operand has elapsed and the TIMEOUT command sequence - if present - has been executed.
 - A REMOVE-CJC-ACTION command has been given.
 - The job is terminated or aborted.
 - The catalog containing a job variable involved is permanently exported.
- The starting time of a CJC command sequence depends on the mode in which the job is when the event “condition satisfied” occurs:

Program mode: A system function called from the program is still being executed.

Command mode: The current command is allowed to terminate.

BREAK/ESCAPE-mode: The start is delayed until a one of the following commands is entered: CANCEL-/RESUME-PROGRAM or CANCEL-/END-/EXIT-/RESUME-PROCEDURE

WAIT mode or
“condition immediately The start occurs immediately.
satisfied”:



If the start of a CJC command sequence is delayed, its definition remains known in the system. Consequently if BREAK/ESCAPE mode lasts long enough, the first “condition satisfied” event can occur in this time and then also the “TIME-LIMIT reached” event. Correspondingly, after BREAK/ESCAPE mode has been exited, if defined the two parts of the CJC command sequence are executed one after the other:

- the part delayed so far for “condition satisfied within the the wait time”
 - and the part for “wait time expired”
- Within the CJC command sequence, only the commands ENTER-JOB, ENTER-PROCEDURE and MODIFY-JV are permitted. ENTER-PROCEDURE is only permitted in a non-S procedure. If an invalid command is specified before the sequence is terminated by END-CJC-ACTION, it is rejected.

- If ENTER-JOB or ENTER-PROCEDURE is the first command in the TIMEOUT command sequence, any job name that is specified must match the name of the command sequence. Example: /.ERROR ENTER-JOB JOB.A, JOB-NAME=ERROR
- In the event of an error in ADD-CJC-ACTION or in the CJC command sequence, procedures apply (interactive/batch): The ADD-CJC-ACTION command rejected and thus is ineffective; control branches to the next of the following commands (after END-CJC-ACTION): SET-JOB-STEP, EXIT-JOB, LOGOFF, CANCEL-PROCEDURE, END-PROCEDURE or EXIT-PROCEDURE. If END-CJC-ACTION is not specified, the procedure terminates abnormally.
- The END-CJC-ACTION command is ignored if the relevant ADD-CJC-ACTION command is missing.
- CJC command sequences are automatically provided by the system with an internal identification which can be used in the REMOVE-CJC-ACTION command.
- As long as an ADD-CJC-ACTION command is effective, no checkpoint (WRCPT macro) can be written. A restart (RESTART-PROGRAM command) terminates all ADD-CJC-ACTION commands still effective.

ADD-CONSOLE-FILTER

Define filter settings for message output

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Operator terminal control
Domain:	CONSOLE-MANAGEMENT
Privileges:	OPERATING
Routing code:	@

Function

The ADD-CONSOLE-FILTER command allows operators to define filter settings (combinations of routing codes and filter levels) for some or all routing codes so as to control message output at operator terminals (consoles) or from authorized user programs. Filter settings the command does not refer to stay as they are. Messages which expect a reply (response messages) and messages sent to a specific recipient (typically command result messages) cannot be suppressed. The ADD-CONSOLE-FILTER command appreciably reduces the volume of message output. It applies only to the operator terminal or authorized user program at or from which it was issued.

Running the command in a user task with the OPERATING privilege

Like any operator terminal, the user task can set filter levels exclusively for itself. The filter levels are effective only when the task is reading from the input stream. When the user task terminates, all the filter levels that it has set are returned to their previous settings.

This function is available regardless of system parameters.

Format

ADD-CONSOLE-FILTER

```
FILTER = *ALL / list-poss(5): <integer 1..5>  
,ROUTING-CODE = *ALL / list-poss(40): <alphanum-name 1..1> / *
```

Operands

FILTER = *ALL / list-poss(5): <integer 1..5>

Defines which filter levels are to be set.

FILTER = *ALL

Sets all filter levels. All command-independent messages which do not expect a reply are suppressed.

FILTER = list-poss(5): <integer 1..5>

Sets the specified filter levels.

ROUTING-CODE = *ALL / list-poss(40): <alphanum-name 1..1> / *

There are 40 routing codes. The filter levels specified in the FILTER operand are set for the routing codes specified here.

ROUTING-CODE = *ALL

All 40 routing codes are affected by the change.

ROUTING-CODE = list-poss(40): <alphanum-name 1..1> / *

The routing codes explicitly specified here are affected by the change.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed successfully
	1	CMD0202	Syntax error in the command
	2	CMD0198	Shutdown in progress
	64	CMD0216	User does not have the necessary privilege
	130	NBR1042	Not enough class 5 memory; cannot set filter levels

Note

Using the OPR parameter service, filter levels for routing codes can be defined for operator terminals and authorized user programs with generated authorization names. If this has not been done, all messages generated between the time the system is initialized and the time the ADD-CONSOLE-FILTER command is issued will be output.

Authorized user programs with dynamic authorization names have no routing codes when first connected. That means that there are no filter levels operative for them at that point.

When an authorized user program with a dynamic authorization name is disconnected, all its filter levels and routing codes are returned to their previous settings.

The REMOVE-CONSOLE-FILTER command can be used to cancel message suppression actions taken with the ADD-CONSOLE-FILTER command.

For further information on routing codes and filter levels see the manual "Introduction to System Administration" [14].

ADD-CRYPTO-PASSWORD

Enter crypto password in the crypto password table of the task

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION HARDWARE-MAINTENANCE

Function

The ADD-CRYPTO-PASSWORD command supplies a job with crypto passwords, which are required to decrypt encrypted files. The crypto password table is set up for the job the first time the ADD-CRYPTO-PASSWORD is issued with the crypto passwords specified there. The table is extended each time the command is called. The crypto password table is automatically deleted at the end of the job. The /REMOVE-CRYPTO-PASSWORD command can be used to delete individual crypto password entries or all the crypto password entries from the table before the end of the job.

When an encrypted file is accessed the system searches the crypto password table for the required crypto password. If it is entered there, access is permitted.

Note

If the rare case of the command being rejected with message DMS066A should occur, this means that the caller has specified a crypto password that, for cryptographic reasons, may not be used. In this case another crypto password must be selected.

Crypto passwords can be up to 8 characters long. Crypto passwords with less than 8 characters are stored left-justified within the system and filled with binary zeros. This means that the two crypto passwords C'ABCD' and C'ABCD_0000' which are specified in the form of a C string are not identical, and owing to the different internal representation two different crypto passwords are calculated by the cryptographic algorithm:

```
/add-crypto-password c'ABCD'          internal: X'C1C2C3C400000000'
/add-crypto-password c'ABCD_0000'    internal: X'C1C2C3C440404040'
```

All crypto passwords and crypto password specifications entered under an active task are counted. Crypto passwords which are entered more than once are ignored and not included in the count.

If the system threshold value for the maximum number of crypto passwords in the crypto password table which is defined with the system parameter PWACTIVE is reached when crypto passwords are entered, the message DMS0691 is output and any further crypto password entry is rejected. This applies for both normal users and systems support. The table entries can be reduced using the /REMOVE-CRYPTO-PASSWORD command and the table made ready to receive new entries.

If the system threshold value for the maximum number of crypto passwords entered in the crypto password table which is defined with the system parameter PWENTERD is reached when crypto passwords are entered under a task, the message DMS0692 is output and any further crypto password entry is rejected. This applies for both normal users and systems support.

Regardless of the setting for the system parameter ENCRYPT, crypto passwords are always stored in encrypted form in the crypto password table.

For information on encrypting files, see also the manual “Introductory Guide to DMS” [13].

REMOTE-FILE-ACCESS

The ADD-CRYPTO-PASSWORD command is automatically forwarded to all RFA partner processes by the requesting job.

Format

ADD-CRYPTO-PASSWORD	Alias: ADCPW
PASSWORD = * SECRET / list-poss(20): <c-string 1..8> / <x-string 1..16>	

Operands

PASSWORD = *SECRET / list-poss(20): <c-string 1..8> / <x-string 1..16>

Passwords which are to be entered in the crypto password table. Up to 20 crypto passwords can be specified in a command. The specification is not case-sensitive.

The operand has the following special characteristics:

- The input field is automatically blanked out in the guided dialog.
- In unguided dialog and foreground procedures, the entry *SECRET or ^, SDF provides a blanked out input field for inputting the password .
- The password entered is not logged.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
	1	CMD0202	Syntactical or semantic error in command
	32	DMS05C7	Unexpected error in DMS
	64	DMS066A	Specified crypto password cannot be used
	64	DMS0691	Crypto password table at maximum size
	64	DMS0692	Maximum number of crypto passwords per task reached
	64	DMS06FF	BCAM connection severed
	130	DMS0594	Not enough virtual memory available

ADD-DEVICE-DEPOT

Define assignment of tape device to depot

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Device management
Domain:	DEVICE
Privileges:	OPERATING
Routing code:	T

This command is available only in conjunction with the MAREN product; the MAREN subsystem must be loaded.

Function

The ADD-DEVICE-DEPOT command is used to notify the device management facility (NDM) of the assignment of physical tape devices (using their mnemonics) to “depots”.

This command cannot be processed successfully unless the following conditions are satisfied:

- there is no allocation to the type of device specified in the command
- there are no allocations to a device type in an existing depot
- the MAREN subsystem is listed in the subsystem catalog

If the first two conditions are not satisfied, message NKG006 is issued to the operator; if the third is not satisfied, message NKG007 is issued.

If the MAREN subsystem is in the subsystem catalog, but is not available at command runtime, any tape devices specified in the command will be permanently inaccessible. Consequently you should not use the command unless the MAREN subsystem is loaded. The system administration must ensure that the depots defined in the MAREN catalog match the depots specified in this command. If this is the case, BS2000 device management in conjunction with MAREN can arrive at a suitable selection of devices for a tape mounting request to be issued to the operator or to a robot-supported cartridge archive system. The definition of depots permits the reservation of devices from a specific device set, e.g. during robot operation (see also the command SECURE-RESOURCE-ALLOCATION, operand DEVICE=(...,LOCATION=...)).

Devices of the same device type assigned to exactly the same depots are administered in “device pools” within NDM. Devices that are not assigned to any depot are classified as “RESTPOOL devices”. The procedure for obtaining depot information is described under “Device management” in the “Introduction to System Administration” [14].

Format

ADD-DEVICE-DEPOT

UNIT = list-poss(10): <alphanum-name 2..2> / <alphanum-name 4..4>

,**LOCATION** = <alphanum-name 1..8>(…)

,**ROUTING-CODE** = *UNCHANGED / <name 1..1>

Operands

UNIT = list-poss(10): <alphanum-name 2..2> / <alphanum-name 4..4>

Mnemonic names of one or more devices to be assigned to a depot. A maximum of 10 devices can be specified per command.

LOCATION = <alphanum-name 1..8>

Name of the depot. Up to 1024 devices can be assigned to a depot.

Note

The depot (LOCATION) is determined by means of MAREN for scratch tape requests. If MAREN is not available, an attempt is made to select a tape device from the RESTPOOL (tape devices which have not been allocated to a depot).

ROUTING-CODE =

Controls message output. If a routing code for which no console is defined in the system is specified, messages are output at the main console.

ROUTING-CODE = *UNCHANGED

The preset value for the routing code remains unchanged.

ROUTING-CODE = <name 1..1>

A defined routing code is specified. This enables the messages necessary for the devices assigned to this depot to be output to a specific console. The preset routing code is T.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
1	0	NKG0008	MAREN subsystem not available
2	0	NKG0007	MAREN subsystem not in subsystem catalog
	64	NKV0004	Command partially processed
	64	NKV0005	Command not processed for an object
	64	NKV0006	Command not processed
	130	NKVT002	Tape monitor unavailable

ADD-FILE-LINK

Create TFT entry and assign values

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT
Routing code:	\$ (with NBCONOPI=N) or. E (with NBCONOPI=Y)

Function

The ADD-FILE-LINK command creates a TFT (Task File Table) entry for the link name specified in the LINK-NAME operand and assigns values to the new entry. If there is already a TFT entry for the specified link name, it is overwritten by the new entry.

The command cannot create new catalog entries or modify the catalog entries of disk files. The command can, however modify the catalog entries of tape files (using the ADD-CATALOG-VOLUME, TAPE-SET-NAME and FILE-SEQUENCE operands, for example). The command cannot be used to modify the storage space allocation of a disk file.

The command cannot modify an existing TST (Tape Set Table) entry; this is the job of the EXTEND-TAPE-SET command.

The command is rejected in the following circumstances:

- A value other than *BY-PROGRAM, *DUMMY or the name of a file which is already in the catalog has been specified in the FILE-NAME operand.
- No disk space has been reserved for the disk file specified in the FILE-NAME operand (though a tape file which is in the catalog may be specified even if the volume table in the catalog entry is empty).

Overview of functions

	Function / Meaning	Level 1 operands	Level 2/3 operands
2-49	Define the file link name for which a TFT entry is to be created	LINK-NAME	
2-50	Name of file/file generation to which the ADD-FILE-LINK command refers	FILE-NAME	
2-50	Access method (ISAM, SAM, BTAM ...)	ACCESS-METHOD	
2-51	OPEN-Mode	OPEN-MODE	
2-52	CLOSE-Mode	CLOSE-MODE	
2-54	Import file attributes from file	DATA-ATTRIBUTES	
2-54	Record format	RECORD-FORMAT	
	Variable record format – feed control	=*VARIABLE	PRINT-CONTROL
2-56	Fixed record format – feed control	=*FIXED	PRINT-CONTROL
2-56	Undefined record format – feed control	=*UNDEFINED	PRINT-CONTROL
2-57	Record length	RECORD-SIZE	
2-58	Block length	BUFFER-LENGTH	
2-60	Block control information	BLOCK-CONTROL-INFO	
2-62	Number of mount requests	NUMBER-OF-PRE MOUNTS	
2-62	Retention period	RETENTION-PERIOD	
2-63	Define volume type	SUPPORT	
2-63	Volume: disk	=*DISK	
2-63	– write access for shared updating		SHARED-UPDATE
2-63	– concurrent writing by jobs from different systems		LOCK-ENVIRONMENT
	– file size restriction		EXCEED-32GB
2-64	– read-after-write checking		WRITE-CHECK
2-65	– performance requirements		IO-ATTRIBUTES
2-65	– performance attribute for file		PERFORMANCE

Table 19: Overview of ADD-FILE-LINK command functions (Part 1 of 3)

	Function / Meaning	Level 1 operands	Level 2/3 operands
2-66	– scope of performance requirements		USAGE
2-66	– ISAM processing specifications		ISAM-ATTRIBUTES
2-67	– key position		KEY-POSITION
2-67	– key length		KEY-LENGTH
2-68	– length of value flag		VALUE-FLAG-LENGTH
2-68	– evaluation of value flags		PROPAGATE-VALUE-FLAG
2-69	– length of logical flags		LOGICAL-FLAG-LENGTH
2-69	– block padding (for sequential processing)		PADDING-FACTOR
2-70	– define pool link name for ISAM pool (NK-ISAM files)		POOL-LINK
2-70	– Size of the file-specific ISAM pool (NK-ISAM files)		POOL-SIZE
2-70	– immediate writing out of updated blocks		WRITE-IMMEDIATE
2-70	– overlapped processing		READ-IN-ADVANCE
2-71	– duplicate keys		DUPLICATE-KEY
2-71	Volume: tape	=*TAPE	
2-71	– list of tapes to process		VOLUME-LIST
2-73	– tape type		DEVICE-TYPE
2-74	– label processing specifications		LABEL-PROCESSING
2-74	– label types		LABEL-TYPE
2-76	– bypass label checking		BYPASS-LABEL-CHECK
2-77	– protection level		PROTECTION-LEVEL
2-78	– tape marks		TAPE-MARK-WRITE
2-78	– define translation table		CODE
2-80	– EBCDIC translation		EBCDIC-TRANSLATION
2-80	– define position in file set		FILE-SEQUENCE
2-81	– automatic checkpoint writing		CHECKPOINT-WRITE
2-81	– file usage on restart		RESTART-USAGE
2-82	– maximum number of blocks per tape		BLOCK-LIMIT

Table 19: Overview of ADD-FILE-LINK command functions (Part 2 of 3)

	Function / Meaning	Level 1 operands	Level 2/3 operands
2-82	– buffer offset		BLOCK-OFFSET
2-83	– buffered/unbuffered processing		TAPE-WRITE
2-83	– overwrite data on remainder of tape		DESTROY-OLD-CONTENTS
2-83	– chained input/output		IO-CHAINING
2-84	– streaming mode		STREAMING-MODE
2-84	Message output on completion of CLOSE processing	FILE-CLOSE-MSG	

Table 19: Overview of ADD-FILE-LINK command functions (Part 3 of 3)

Format

(Part 1 of 4)

ADD-FILE-LINK	Alias: ADFL
<p>LINK-NAME = <filename 1..8 without-gen></p> <p>,FILE-NAME = <u>*BY-PROGRAM</u> / *DUMMY / <filename 1..54></p> <p>,ACCESS-METHOD = <u>*BY-DATA-ATTRIBUTES</u> / *BY-PROGRAM / *BY-CATALOG / *SAM / *ISAM / *BTAM / *UPAM</p> <p>,OPEN-MODE = <u>*BY-PROGRAM</u> / *INPUT / *OUTPUT / *EXTEND / *REVERSE / *UPDATE / *OUTIN / *INOUT / *SINOUT</p> <p>,CLOSE-MODE = <u>*BY-PROGRAM</u> / *REWIND / *REPOS / *UNLOAD / *LEAVE / *INVALIDATE / *KEEP-DATA-IN-CACHE</p> <p>,DATA-ATTRIBUTES = *STD / *FROM-FILE(...) / *BY-CATALOG</p> <p> *FROM-FILE(...) FILE-NAME = <filename 1..54></p> <p>,RECORD-FORMAT = <u>*BY-DATA-ATTRIBUTES</u> / *BY-PROGRAM / *BY-CATALOG / *VARIABLE(...) / *FIXED(...) / *UNDEFINED(...)</p> <p> *VARIABLE(...) PRINT-CONTROL = *NONE / *ASA / *EBCDIC</p> <p> *FIXED(...) PRINT-CONTROL = *NONE / *ASA / *EBCDIC</p> <p> *UNDEFINED(...) PRINT-CONTROL = *NONE / *ASA / *EBCDIC</p> <p>,RECORD-SIZE = <u>*BY-DATA-ATTRIBUTES</u> / *BY-PROGRAM / *BY-CATALOG / <integer 0..32768></p>	

```

, BUFFER-LENGTH = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / *STD(...) /
    <integer 1..32768>

    *STD(...)
    |   SIZE = 1 / <integer 1..16>

, BLOCK-CONTROL-INFO = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / *NO /
    *WITHIN-DATA-BLOCK / *WITHIN-DATA-2K-BLOCK /
    *WITHIN-DATA-4K-BLOCK / *PAMKEY

, NUMBER-OF-PREMOUNTS = *STD / <integer 0..255>

, RETENTION-PERIOD = *BY-PROGRAM / <integer 0..32767 days>

, SUPPORT = *NONE / list-poss(2): *DISK(...) / *TAPE(...)

*DISK(...)
    SHARED-UPDATE = *BY-PROGRAM / *NO / *YES / *WEAK

, LOCK-ENVIRONMENT = *BY-PROGRAM / *HOST-SYSTEM / *XCS

, EXCEED-32GB = *BY-PROGRAM / *FORBIDDEN / *ALLOWED

, WRITE-CHECK = *BY-PROGRAM / *NO / *YES

, IO-ATTRIBUTES = *BY-PROGRAM / [*PARAMETERS](...)
    [*PARAMETERS](...)
    |   PERFORMANCE = *BY-PROGRAM / *BY-CATALOG / *STD / *HIGH / *VERY-HIGH /
    |   *USER-MAXIMUM
    |   , USAGE = *BY-PROGRAM / *BY-CATALOG / *READ-WRITE / *WRITE / *READ

, ISAM-ATTRIBUTES = *BY-PROGRAM / [*PARAMETERS](...)
    [*PARAMETERS](...)
    |   KEY-POSITION = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG /
    |   <integer 1..32767>
    |   , KEY-LENGTH = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG /
    |   <integer 1..255>
    |   , VALUE-FLAG-LENGTH = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG /
    |   <integer 0..255>
    |   , PROPAGATE-VALUE-FLAG = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *MINIMUM /
    |   MAXIMUM / *BY-CATALOG
    |   , LOGICAL-FLAG-LENGTH = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG /
    |   <integer 0..255>
    |   , PADDING-FACTOR = *BY-PROGRAM / <integer 0..99>
    |   , POOL-LINK = *BY-PROGRAM / <name 1..8>
    |   , POOL-SIZE = *BY-PROGRAM / <integer 128..1048576 2Kbyte>

```

	<p>,WRITE-IMMEDIATE = <u>*BY-PROGRAM</u> / *NO / *YES</p> <p>,READ-IN-ADVANCE = <u>*BY-PROGRAM</u> / *YES / *NO</p> <p>,DUPLICATE-KEY = <u>*BY-PROGRAM</u> / *YES / *NO</p>
*TAPE(...)	<p>VOLUME-LIST = <u>*CATALOG</u> (...) / *TEMPORARY(...) / *TAPE-SET(...)</p> <p>*CATALOG(...)</p> <p> ADD-CATALOG-VOLUME = <u>*NONE</u> / *ANY(...) / list-poss(255): <alphanum-name 1..6></p> <p> *ANY(...)</p> <p> NUMBER-OF-DEVICES = <u>1</u> / <integer 1..9></p> <p> ,VOL-SEQUENCE-NUMBER = <u>*NONE</u> / *FROM-START-POSITION(...) /</p> <p> list-poss(255): <integer 1..255></p> <p> *FROM-START-POSITION(...)</p> <p> START-POSITION = <integer 1..255></p> <p>*TEMPORARY(...)</p> <p> PROCESS-VOLUME = list-poss(255): <alphanum-name 1..6></p> <p>*TAPE-SET(...)</p> <p> TAPE-SET-NAME = <alphanum-name 1..4></p> <p> ,FILE-SET-IDENTIFIER = <u>*BY-TAPE-SET</u> / <alphanum-name 1..6></p> <p>,DEVICE-TYPE = <u>*ANY</u> / <device></p> <p>,LABEL-PROCESSING = <u>*BY-PROGRAM</u> / [*PARAMETERS](...)</p> <p>[*PARAMETERS](...)</p> <p> LABEL-TYPE = <u>*BY-DATA-ATTRIBUTES</u> / *BY-PROGRAM / *NO / *NON-STD / *STD(...)</p> <p> *STD(...)</p> <p> DIN-REVISION-NUMBER = <u>*BY-PROGRAM</u> / <integer 0..3></p> <p> ,BYPASS-LABEL-CHECK = <u>*BY-PROGRAM</u> / *NO-POSITIONING /</p> <p> *ABSOLUTE-POSITIONING(...) / *FORWARD-POSITIONING(...) /</p> <p> *BACKWARD-POSITIONING(...)</p> <p> *ABSOLUTE-POSITIONING(...)</p> <p> TAPE-MARK-NUMBER = <integer 0..32767></p> <p> *FORWARD-POSITIONING(...)</p> <p> NUMBER-OF-TAPE-MARKS = <integer 0..127></p> <p> *BACKWARD-POSITIONING(...)</p> <p> NUMBER-OF-TAPE-MARKS = <integer 0..127></p>


```

,PROTECTION-LEVEL = *BY-PROGRAM / *LOW(...) / *HIGH(...)
    *LOW(...)
    |   OVERWRITE-PROTECTION = *NO / YES
    *HIGH(...)
    |   OVERWRITE-PROTECTION = *NO / YES
,TAPE-MARK-WRITE = *BY-PROGRAM / *YES

,CODE = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / *EBCDIC / *ISO7 / *ISO7D /
    *OWN

,EBCDIC-TRANSLATION = *BY-PROGRAM / *YES / *NO

,FILE-SEQUENCE = *BY-PROGRAM / *BY-CATALOG / *UNKNOWN / *NEW / <integer 0..9999>

,CHECKPOINT-WRITE = *BY-PROGRAM / [*PARAMETERS](...)
    [*PARAMETERS](...)
    |   CHKPT-AT-BLOCK-LIMIT = *BY-PROGRAM / *YES
    |   ,CHKPT-AT-FORCED-EOV = *BY-PROGRAM / *YES

,RESTART-USAGE = *BY-PROGRAM / *DUMMY

,BLOCK-LIMIT = *BY-PROGRAM / <integer 1..999999>

,BLOCK-OFFSET = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / *BY-HDR2 /
    <integer 0..99 byte>

,TAPE-WRITE = *BY-PROGRAM / *DEVICE-BUFFER / *IMMEDIATE

,DESTROY-OLD-CONTENTS = *BY-CATALOG / *NO / *YES

,IO-CHAINING = *BY-PROGRAM / <integer 1..16>

,STREAMING-MODE = *BY-PROGRAM / *YES

,FILE-CLOSE-MSG = *STD / *NO / *YES

```

Operands

LINK-NAME = <filename 1..8 without-gen>

A link name under which programs can open the file.

The file link name/TFT links the program and file together. A TFT entry is set up for the file link name specified here, and the remaining operands are evaluated and the values copied into this TFT entry. If there is already an entry in the TFT under the specified link name, it is overwritten by the new one.

If the old TFT entry was locked by a LOCK-FILE-LINK command, the new entry is also locked.

The old volume and device reservations are canceled; tape devices remain available to the job, but can be released by a SECURE-RESOURCE-ALLOCATION command.

A TSET name cannot be used as a file link name.

FILE-NAME = *BY-PROGRAM / *DUMMY / <filename 1..54>

Identifies the file to which the ADD-FILE-LINK command refers; FILE-NAME must not be a file generation group. The file specified by FILE-NAME must already be cataloged. Unless it is a tape file, it must already have been allocated storage space. Otherwise the ADD-FILE-LINK command is rejected.

FILE-NAME = *BY-PROGRAM

The file name specified in the program is used.

FILE-NAME = *DUMMY

Creates a TFT entry for a DUMMY file under the specified link name. No devices, volumes or storage space are allocated.

DUMMY files are typically used during program debugging to simulate I/O transactions, or during a restart with the RESTART-PROGRAM command to replace files which are no longer required for processing but without which the program will not run (see the RESTART-PROGRAM command description).

DUMMY file as input file: if an attempt is made to read the file, EOF processing is initiated, which means that the program acts as though the file had already been read. DUMMY file as output file: if an attempt is made to write to the file, the data is transferred to the program's buffer areas, but output to a volume is suppressed.

FILE-NAME = <filename 1..54>

This is the file the ADD-FILE-LINK command refers to.

ACCESS-METHOD = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / *SAM / *ISAM / *BTAM / *UPAM

Specifies the access method for file processing.

Access methods are described in the "Introductory Guide to DMS" [13].

ACCESS-METHOD = *BY-DATA-ATTRIBUTES

The access method of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, ACCESS-METHOD defaults to *BY-PROGRAM.

ACCESS-METHOD = *BY-PROGRAM

The access method specified in the program is used.

ACCESS-METHOD = *BY-CATALOG

The access method entered in the catalog is used (see SHOW-FILE-ATTRIBUTES command, *FILE-STRUCT* output field).

ACCESS-METHOD = *SAM

The file specified in the FILE-NAME operand is to be processed using the SAM access method. Both disk and tape files can be processed.

ACCESS-METHOD = *ISAM

The file specified in the FILE-NAME operand is to be processed using the ISAM access method. Only disk files can be processed.

ACCESS-METHOD = *BTAM

The file specified in the FILE-NAME operand is to be processed using the BTAM access method. Only tape files can be processed. BTAM processing is not supported by all programming languages. BTAM-specific operands: IO-CHAINING, OPEN-MODE=*SINOUT

ACCESS-METHOD = *UPAM

The file specified in the FILE-NAME operand is to be processed using the UPAM access method. Both disk and tape files can be processed.

OPEN-MODE = *BY-PROGRAM / *INPUT / *OUTPUT / *EXTEND / *REVERSE / *UPDATE / *OUTIN / *INOUT / *SINOUT

Specifies the OPEN mode for the file. This setting may be overridden by the OPEN statement in the program when the file is actually opened. The following table shows which OPEN modes are permissible for the various access methods (D,T,X) and which are not permissible (-).

OPEN-MODE=	Access method			
	ISAM	BTAM	SAM	UPAM
*INPUT	D	T	X	X
*EXTEND	D	-	X	-
*INOUT	D	T	-	X
*OUTIN	D	T	-	X
*OUTPUT	D	T	X	-
*REVERSE	-	T	X	-
*SINOUT	-	T	-	-
*UPDATE	-	-	D	-

Key:

P: Disk files (SUPPORT=*DISK)

X: Disk and tape files

B: Tape files (SUPPORT=*TAPE)

-: OPEN mode not supported

Table 20: OPEN modes and access methods

OPEN-MODE = *BY-PROGRAM

The OPEN mode specified in the program is used.

OPEN-MODE = *INPUT

The file specified under FILE-NAME is used as an input file (and must therefore already exist).

OPEN-MODE = *OUTPUT

The file is created or, if it already exists, overwritten from start of file. If it is a tape file, a label is created.

OPEN-MODE = *EXTEND

Extends an existing file, i.e. adds blocks to the end of the file, or overwrites the file from a specific point onwards; only sequential writes are allowed. For tape files, labels are generated as appropriate to the LABEL-TYPE specification.

OPEN-MODE = *REVERSE

The file, which is opened as an input file for sequential reading, must already exist. It is processed in reverse order, i.e. from end of file to start of file. With tape files, automatic tape swapping is not possible. On completion of OPEN processing, the tape is positioned at the end of the file section.

OPEN-MODE = *UPDATE

For SAM disk files only: the records in the file can be modified using the GET macro followed by the PUTX macro (possible only in move mode).

OPEN-MODE = *OUTIN

The file is created or, if it already exists, overwritten from start of file. Both (nonsequential) writes and reads are allowed. Labels are written in the case of tape files.

OPEN-MODE = *INOUT

Opens an existing file for nonsequential processing; both reads and writes are allowed. Tapes are positioned to start of tape on completion of OPEN processing; no labels are written.

OPEN-MODE = *SINOUT

For tape files only: the file must exist, and the tape must not be positioned at start of tape; data blocks can be read or written, and no label processing is performed. Unlike *INOUT, this mode does not position the tape.

CLOSE-MODE = *BY-PROGRAM / *REWIND / *REPOS / *UNLOAD / *LEAVE / *INVALIDATE / *KEEP-DATA-IN-CACHE

Specifies the CLOSE mode to be used to close the file. This setting may be overridden by a CLOSE macro when the file is actually closed.

See the CLOSE macro description in the “DMS Macros” manual [12]).

CLOSE-MODE = *BY-PROGRAM

The CLOSE mode is defined by the CLOSE macro in the program.

CLOSE-MODE = *REWIND

For tape files only: regardless of the LABEL-TYPE specification, the tape is rewound to the start once the file has been closed.

CLOSE-MODE = *REPOS

For tape files only: the tape is repositioned to the start of the current file section once the file has been closed.

CLOSE-MODE = *UNLOAD

For tape files only: the tape is rewound, unloaded and released once the file has been closed.

CLOSE-MODE = *LEAVE

For tape files only: depending on the LABEL-TYPE specification, the tape is positioned to logical end of file once the file has been closed.

CLOSE-MODE = *INVALIDATE

Cached writes for the file are not written out to disk. All the data in the cache is marked as invalid.

This value is useful for files for which (cached) writes are no longer needed once they have been closed. This avoids wasting time writing unsaved cache data out to disk.

Once a file has been closed with CLOSE-MODE=*INVALIDATE, any writes for it are lost. Thereafter the file must under no circumstances be opened with OPEN-MODE= *INPUT or *OUTPUT.

CLOSE-MODE = *KEEP-DATA-IN-CACHE

Cached writes for the file are not written out to disk. All the data in the cache is retained there.

This value is useful for follow-up processing on the file, since no time is wasted writing the cache data out to disk when the file is closed. The cache data is not invalidated, and the next time the file is opened, read accesses can again be serviced immediately from the cache without the need to read the data back into it. This special variant of PFA caching is also known as the HIPERBATCH mechanism (**H**igh-**P**erformance **B**atch Processing; see also the "Introduction to System Administration" [14]).

DATA-ATTRIBUTES = *STD / *FROM-FILE(...) / *BY-CATALOG

Specifies whether the *BY-DATA-ATTRIBUTES defaults of the following operands are to be the same as the corresponding file attributes of a cataloged file. The operands which support the importing of file attributes are: ACCESS-METHOD, RECORD-FORMAT, RECORD-SIZE, BUFFER-LENGTH, BLOCK-CONTROL-INFO, KEY-POSITION, KEY-LENGTH, LOGICAL-FLAG-LENGTH, VALUE-FLAG-LENGTH, PROPAGATE-VALUE-FLAG, BLOCK-OFFSET, LABEL-TAPE and CODE.

DATA-ATTRIBUTES = *STD

The *BY-DATA-ATTRIBUTES defaults are the same as the corresponding *BY-PROGRAM operand values.

DATA-ATTRIBUTES = *FROM-FILE(...)

Operands for which *BY-DATA-ATTRIBUTES is specified are assigned the operand value corresponding to the matching file attribute of the named file. If the operand value thus imported initiates a structure, the values of the lower-level operands are likewise set to match the equivalent file attributes. This affects the PRINT-CONTROL, SIZE and DIN-REVISION-NUMBER operands.

FILE-NAME = <filename 1..54>

Name of the file from which the corresponding file attributes are to be imported. The caller must have authorization to read the file's catalog entry with SHOW-FILE-ATTRIBUTES. The file must be cataloged on the same pubset as the file to which the ADD-FILE-LINK command refers.

DATA-ATTRIBUTES = *BY-CATALOG

The operand values are taken over as with DATA-ATTRIBUTES=*FROM-FILE(...), namely from the file to which the ADD-FILE-LINK command refers.

RECORD-FORMAT = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / *VARIABLE(...) / *FIXED(...) / *UNDEFINED(...)

Specifies the record format of the file identified by FILE-NAME. The block oriented access method UPAM ignores the RECORD-FORMAT argument. BTAM is also a block oriented access method, but it does accept specifications in the RECORD-FORMAT operand.

*UNDEFINED is treated as if it were *VARIABLE.

The SAM and ISAM access methods interpret the record format argument, except that *UNDEFINED is not allowed with ISAM.

Record formats are described in full in the "Introductory Guide to DMS" [13].

For the relationship between the RECORD-FORMAT and RECORD-SIZE arguments see the description of the RECORD-SIZE operand.

RECORD-FORMAT = *BY-DATA-ATTRIBUTES

The record format of the file specified in the DATA-ATTRIBUTES operand is used. The setting of the print control byte (PRINT-CONTROL) is also copied. If DATA-ATTRIBUTES is *STD, RECORD-FORMAT defaults to *BY-PROGRAM.

RECORD-FORMAT = *BY-PROGRAM

The record format selected in the program is used.

RECORD-FORMAT = *BY-CATALOG

The record format set in the catalog is used.

RECORD-FORMAT = *VARIABLE(...)

The file specified by FILE-NAME 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, where bytes 1-2 contain the record length in binary. The bytes 3-4 are used by the system. For input files, the record length field is specified by the system, for output files, this must be done by the user.

PRINT-CONTROL = *NONE / *ASA / *EBCDIC

Identifies the feed control character (see the PRINT-DOCUMENT command, operand DOCUMENT-FORMAT=*TEXT(LINE-SPACING=...)).

PRINT-CONTROL = *NONE

The file specified by FILE-NAME is not a print file; it therefore does not contain any print control characters and when printed should not be subject to control character evaluation.

PRINT-CONTROL = *ASA

The first data byte in each record is to be interpreted as an ASA feed control character for the printer (print with LINE-SPACING=*BY-ASA-CONTROL in the PRINT-DOCUMENT command).

PRINT-CONTROL = *EBCDIC

The first data byte in each record is to be interpreted as an EBCDIC feed control character for the printer (print with LINE-SPACING=*BY-EBCDIC-CONTROL in the PRINT-DOCUMENT command). With ISAM files the ISAM index is taken into account.

RECORD-FORMAT = *FIXED(...)

The file specified by FILE-NAME consists of fixed-length records, which means that the user does not need to allow for record length and control fields. All the records in the file are of the same length, this length being defined in the RECORD-SIZE operand.

PRINT-CONTROL = *NONE / *ASA / *EBCDIC

Identifies the feed control character (see the PRINT-DOCUMENT command, operand DOCUMENT-FORMAT=*TEXT(LINE-SPACING=...)).

PRINT-CONTROL = *NONE

The file specified by FILE-NAME is not a print file; it therefore does not contain any print control characters and when printed should not be subject to control character evaluation.

PRINT-CONTROL = *ASA

The first data byte in each record is to be interpreted as an ASA feed control character for the printer (print with LINE-SPACING=*BY-ASA-CONTROL in the PRINT-DOCUMENT command).

PRINT-CONTROL = *EBCDIC

The first data byte in each record is to be interpreted as an EBCDIC feed control character for the printer (print with LINE-SPACING=*BY-EBCDIC-CONTROL in the PRINT-DOCUMENT command). With ISAM files the ISAM index is taken into account.

RECORD-FORMAT = *UNDEFINED(...)

The file specified by FILE-NAME consists of records of "undefined" length; each data block contains a single record, the length of which is passed in a register; the value in this register is set by the system on input and by the user on output (see the RECORD-SIZE operand). RECORD-FORMAT=*UNDEFINED converts the specification LABEL-TYPE=*STD(DIN-REVISION-NUMBER=3) to LABEL-TYPE=*STD(DIN-REVISION-NUMBER=2).

PRINT-CONTROL = *NONE / *ASA / *EBCDIC

Identifies the feed control character (see the PRINT-DOCUMENT command, operand DOCUMENT-FORMAT=*TEXT(LINE-SPACING=...)).

PRINT-CONTROL = *NONE

The file specified by FILE-NAME is not a print file; it therefore does not contain any print control characters and when printed should not be subject to control character evaluation.

PRINT-CONTROL = *ASA

The first data byte in each record is to be interpreted as an ASA feed control character for the printer (print with LINE-SPACING=*BY-ASA-CONTROL in the PRINT-DOCUMENT command).

PRINT-CONTROL = *EBCDIC

The first data byte in each record is to be interpreted as an EBCDIC feed control character for the printer (print with LINE-SPACING=*BY-EBCDIC-CONTROL in the PRINT-DOCUMENT command). With ISAM files the ISAM index is taken into account.

RECORD-SIZE = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / <integer 0..32768>

With RECORD-FORMAT=*FIXED, the RECORD-SIZE operand specifies the common length of all records in bytes.

With RECORD-FORMAT=*VARIABLE, the value of RECORD-SIZE is ignored. This does not apply to ISAM file reads, where the value in RECORD-SIZE is interpreted as a maximum record length (in bytes); with RECORD-SIZE=0, the length of a logical block is used as the maximum record length. If the program reads a record which is longer than the maximum record length, only the maximum length as defined by RECORD-SIZE is transferred and error handling is initiated.

With RECORD-FORMAT=*UNDEFINED, the value of RECORD-SIZE is interpreted as the number of a general-purpose register ($2 \leq \text{register} \leq 12$) containing the current record length. On input, the system supplies the register with the record length; on output, the user must supply the value.

With tape files there is a significant interaction with the CODE and LABEL-PROCESSING operands: when used together with CODE=*EBCDIC or LABEL-PROCESSING= *PARAMETERS LABEL=*STD (DIN-REVISION-NUMBER > 1), the RECORD-SIZE operand must have a value ≤ 9999 (international standard).

RECORD-SIZE = *BY-DATA-ATTRIBUTES

The value of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, RECORD-SIZE defaults to *BY-PROGRAM.

RECORD-SIZE = *BY-PROGRAM

The value specified in the program is set.

RECORD-SIZE = *BY-CATALOG

The record format set in the catalog is used.

RECORD-SIZE = <integer 0..32768>

Specifies the maximum record length in bytes. Note with NK-ISAM files that overflow blocks are generated if the maximum record length is utilized.

BUFFER-LENGTH = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / <integer 1..32768> / *STD(...)

Defines the logical block length. A logical block is the unit of information transmitted to and from the I/O devices as seen by the user calling the access methods.

In the case of disk files there are interactions with space allocation (CREATE-FILE/MODIFY-FILE-ATTRIBUTES command, SPACE operand) and record length (ADD-FILE-LINK command, RECORD-SIZE operand). In the case of tape files there is an interaction with the label attributes (ADD-FILE-LINK command, LABEL-TYPE operand). See also [table 21 on page 2-59](#) and [table 23 on page 2-75](#).

Disk files/tape files with standard blocks: data blocks may consist of a number of PAM pages. The system automatically links together the PAM pages which form a transmission unit.

BUFFER-LENGTH = *BY-DATA-ATTRIBUTES

The logical block length of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, BUFFER-LENGTH defaults to *BY-PROGRAM. If the file consists of standard blocks (BUFFER-LENGTH=*STD), the number of PAM pages (SIZE operand) is also copied from the file.

BUFFER-LENGTH = *BY-PROGRAM

The value set in the program is used.

BUFFER-LENGTH = *BY-CATALOG

The value set in the catalog is used.

BUFFER-LENGTH = <integer 1..32768>

For tape files only: defines the block length in bytes and specifies that the file consists of nonstandard blocks (i.e. blocks which are not PAM blocks). Each nonstandard block is also a logical block. With RECORD-FORMAT=*FIXED, all nonstandard blocks in the file are of the length specified here (buffer offset not included; see BLOCK-OFFSET operand). With RECORD-FORMAT=*VARIABLE/*UNDEFINED, the nonstandard blocks may be of differing lengths, with the specified block length defining the upper limit (buffer offset included).

The ACCESS-METHOD and IO-CHAINING operand settings must also be taken into account.

BUFFER-LENGTH = *STD(...)

The file consists of standard PAM pages.

SIZE = 1 / <integer 1..16>

For K files: each logical block consists of the number of PAM blocks specified here.

For NK files: each logical block consists of the number of 2048-byte data fields specified here. With NK4 volumes (NK4 files) this number must be even; in other words, the logical block length is a multiple of 4K.

For tape files: if CODE is anything other than EBCDIC, or if BLOCK-CONTROL-INFO is set to *WITHIN-DATA-BLOCK or *NO, STD block specifications are converted to nonstandard block specifications.

RECORD-FORMAT operand	Effects
RECORD-FORMAT=*FIXED	BUFFER-LENGTH specifies the block length including the length of the buffer offset (see BLOCK-OFFSET operand); all blocks are of the same length
RECORD-FORMAT= *VARIABLE/*UNDEFINED	BUFFER-LENGTH specifies the maximum block length including the length of the buffer offset (see the BLOCK-OFFSET operand), so the block length (like the record length) is variable If RECORD-FORMAT=VARIABLE and CODE= * EBCDIC or LABEL=*STD(DIN-REV-NUM=n) (n> 1), BUFFER-LENGTH must be less than 10000 (internal conversion to record format D)

Table 21: Tape files: block length and record format

ACCESS-METHOD operand	Values allowed for BUFFER-LENGTH
SAM / BTAM UPAM	$1 \leq n \leq 32768$

Table 22: Tape files: block length and record format

BLOCK-CONTROL-INFO = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / *NO / *WITHIN-DATA-BLOCK / *WITHIN-DATA-2K-BLOCK / *WITHIN-DATA-4K-BLOCK / *PAMKEY

Specifies the location of the block control field containing the block control information used for logical block management. K files have the block format PAMKEY and can only be stored on K volumes. NK files can have any of the following formats: NO, WITHIN-DATA-BLOCK, WITHIN-DATA-2K-BLOCK and WITHIN-DATA-4K-BLOCK. They can be stored on N or NK volumes. NK files are subdivided into:

- NK2 files, which may have the following structural attributes in the catalog entry:
 - *FILE-STRUCT = ISAM* and *BLK-CONTR = DATA (2K)*
 - *FILE-STRUCT = SAM* and *BLK-CONTR = DATA* and *BUF-LEN = STD(n)*, where n is an odd number
 - *FILE-STRUCT = PAM* and *BLK-CONTR = DATA* or *NO* and *BUF-LEN = STD(n)*, where n is an odd number
 NK2 files **cannot** be stored on NK4 pubsets.
- NK4 files, which may have the following structural attributes in the catalog entry:
 - *FILE-STRUCT = ISAM* and *BLK-CONTR = DATA (4K)*
 - *FILE-STRUCT = SAM* and *BLK-CONTR = DATA* and *BUF-LEN = STD(n)*, where n is an even number
 - *FILE-STRUCT = PAM* and *BLK-CONTR = DATA* or *NO* and *BUF-LEN = STD(n)*, where n is an even number

BLOCK-CONTROL-INFO = *BY-DATA-ATTRIBUTES

The value of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, BLOCK-CONTROL-INFO defaults to *BY-PROGRAM.

BLOCK-CONTROL-INFO = *BY-PROGRAM

The value specified in the program is used.

BLOCK-CONTROL-INFO = *BY-CATALOG

The value set in the catalog is used.

BLOCK-CONTROL-INFO = *NO

The block format does not include a PAM key, i.e. the system does not store any block control information in PAM keys. This block format is allowed for NK files only. There is available only for PAM files and SAM tape files. For SAM disk files and ISAM, *NO is treated in the same way as *WITHIN-DATA-BLOCK. Functions which used to be linked to data in the PAM key are no longer supported (this typically applies to partial backups using ARCHIVE).

BLOCK-CONTROL-INFO = *WITHIN-DATA-BLOCK

The block format does not include a PAM key, i.e. the system does not store any block control information in PAM keys. This block format is allowed for NK files only. The block control information is within the first twelve bytes of each logical block, except in NK-ISAM files. Incompatibilities arise if the maximum record length (16 PAM blocks = 32K bytes) is used up, as there is no room left for the block control information.

Special features of NK-ISAM files:

There are two block formats:

- 2K format: The block control information is stored at the start of each 2K block (one PAM page). The 2K format can be selected explicitly using BLOCK-CONTROL-INFO=*WITHIN-DATA-2K-BLOCK.
- 4K format: The block control information is stored at the start of each 4K block (two PAM pages). The 4K format can be selected explicitly using BLOCK-CONTROL-INFO=*WITHIN-DATA-4K-BLOCK.

The block format used to create an NK-ISAM file (OPEN=*OUTPUT/*OUTIN) is governed by the format of the volume on which the file is stored:

K pubset/private disk	}	K format (NK2 ISAM file): catalog entry where BLK-CONTR = DATA (2K)
NK2(6K) pubset/private disk		
NK2(8K) pubset NK2(64K) pubset		
NK4(8K) pubset	}	4K format (NK4 ISAM file): catalog entry where BLK-CONTR = DATA (4K)
NK4(64K) pubset		

Note that it is not possible to create the NK-ISAM file in 4K format unless the logical block length is a multiple of 4K. If this is not the case, any attempt to open the file is rejected as an error.

An existing NK-ISAM can be opened regardless of the block format.

BLOCK-CONTROL-INFO = *WITHIN-DATA-2K-BLOCK

*This is a variant of *WITHIN-DATA-BLOCK designed for use with NK-ISAM files.* When an NK-ISAM file is created (OPEN=*OUTPUT/*OUTIN), 2K format is generated. Any attempt to open the file for an NK4 volume is rejected as an error. An existing NK-ISAM file can be opened only if it was created in 2K format.

BLOCK-CONTROL-INFO = *WITHIN-DATA-4K-BLOCK

*This is a variant of *WITHIN-DATA-BLOCK designed for use with NK-ISAM files.*

When an NK-ISAM file is created (OPEN=*OUTPUT/*OUTIN), 4K format is generated. The logical block length must be a multiple of 4K. If this is not the case, any attempt to open the file is rejected as an error. An existing NK-ISAM file can be opened only if it was created in 4K format.

BLOCK-CONTROL-INFO = *PAMKEY

This values is possible for K files only.

A block format using PAM keys which is supported by all access methods. The block control information is stored in the PAM key. A file may be created in PAMKEY format provided that the volume supports PAMKEY recording (K pubset or K private disk).

NUMBER-OF-PREMOUNTS = *STD / <integer 0..255>

Number of volumes to be requested for premounting.

NUMBER-OF-PREMOUNTS = *STD

Disk files: premounts all disks containing an extent of the file.

Tape files: premounts a single tape.

NUMBER-OF-PREMOUNTS = <integer 0..255>

Specifies the number of volumes to premount.

RETENTION-PERIOD = *BY-PROGRAM / <integer 0..32767 days>

This operand allows users to define a retention period for their files. When the output file is closed, the date on which the retention period expires is calculated from the current data and the length of the retention period. This date is added to the catalog entry as the expiration date. No write access (updating, deletion) to this file is possible until the retention date has been reached.

RETENTION-PERIOD = *BY-PROGRAM

The value specified in the program is used.

RETENTION-PERIOD = <integer 0..32767 days>

The RETENTION-PERIOD operand is effective only if a TFT entry is created using the LINK-NAME operand, and file identified by the FILE-NAME operand is then opened. The value of RETENTION-PERIOD defines the length of the retention period in days. Expiration of the retention period does not mean that the file will automatically be destroyed, merely that write access will then be granted again. RETENTION-PERIOD=0 means that there is no retention period and the file can be updated or deleted at any time. A retention period can also be set with the MODIFY-FILE-ATTRIBUTES command; a retention period specified with that command is added to the catalog entry immediately; but note that for tape files the retention period must be defined before the file is opened for the first time.

SUPPORT = *NONE / list-poss(2): *DISK(...) / *TAPE(...)

Type of volume on which the file is located.

SUPPORT = *NONE

No device specification.

SUPPORT = *DISK(...)

The file to be processed is a disk file.

SHARED-UPDATE = *BY-PROGRAM / *NO / *YES / *WEAK

For ISAM and UPAM disk files only: defines whether write access to the file is to be possible if other jobs open it at the same time (shared updating).

SHARED-UPDATE = *BY-PROGRAM

The value specified in the program is used.

SHARED-UPDATE = *NO

Once one job has opened the file in an OPEN mode other than INPUT, it is locked for all other jobs. Thus shared updating is not possible. The contents of the file remain constant for the duration of file processing. Concurrent access to the file by a number of jobs is not possible unless the file is opened as an input file, i.e. with OPEN INPUT, in all jobs. If the file has already been opened in INPUT mode, any attempt to open it in any other mode is rejected.

SHARED-UPDATE = *YES

For ISAM and PAM files only: the file can be edited concurrently by a number of jobs; but SHARED-UPDATE=*YES must be set in all jobs. With UPAM, users can protect data blocks they are processing against access by other jobs. With ISAM, these locks are set automatically by the system as required. With NK-ISAM, files opened for shared updating must be processed in cross-task user ISAM pools. With ISAM files the WRITE-IMMEDIATE function is enabled concurrently (WRITE-IMMEDIATE operand set to *YES).

SHARED-UPDATE = *WEAK

For UPAM processing only: guarantees write but not read protection. Only one job can open the file for writing. Other jobs can use it concurrently as an input file, but allowance has to be made for the fact that the contents of the file may change.

LOCK-ENVIRONMENT = *BY-PROGRAM / *HOST-SYSTEM / *XCS

Specifies whether the file can be open for concurrent writing by jobs from different systems, depending on the open mode (OPEN-MODE) and the shared update mode setting (SHARED-UPDATE).

LOCK-ENVIRONMENT = *BY-PROGRAM

The setting specified in the program is used.

LOCK-ENVIRONMENT = *HOST-SYSTEM

The file cannot be open for concurrent writing with SHARED-UPDATE=*YES by jobs from different systems.

LOCK-ENVIRONMENT = *XCS

The file can be open for concurrent writing with SHARED-UPDATE=*YES by jobs from different systems in an XCS network.

EXCEED-32GB = *BY-PROGRAM / *FORBIDDEN / *ALLOWED

Specifies whether the file size may exceed 32 GB during processing.

EXCEED-32GB = *FORBIDDEN

The file may grow to a maximum of 32 GB.

EXCEED-32GB = *ALLOWED

The file may grow beyond 32 GB.

WRITE-CHECK = *BY-PROGRAM / *NO / *YES

Specifies whether records written out to disk are to be checked immediately (read-after-write check) so that recording errors are detected promptly. The WRITE-CHECK specification is not added to the catalog entry, so it must be repeated beforehand each time the file is processed or opened.

Read-after-write checking: checking for recording errors (→ error recovery actions). If the error is unrecoverable, control is passed to the EXLST exit ERRADR. Read-after-write checking slows down performance considerably on account of the extra disk rotations.

WRITE-CHECK = *BY-PROGRAM

The setting specified in the program is used.

WRITE-CHECK = *NO

No read-after-write checking is done.

WRITE-CHECK = *YES

Read-after-write checking is done.

IO-ATTRIBUTES = *BY-PROGRAM / *PARAMETERS(...)

Enables the user to indicate performance requirements to the system (for I/O operations). The highest permissible performance attribute is defined in the user entry. The cache medium defined for the associated pubset governs whether and to what extent performance requirements are implemented (see the output of the SHOW-MASTER-CATALOG-ENTRY command). Read and write accesses are handled by high-speed buffers. This reduces the number of disk accesses and the average access time.

IO-ATTRIBUTES = *BY-PROGRAM

The value specified in the program is used (see the FCB macro description in the “DMS Macros” manual [12]).

IO-ATTRIBUTES = *PARAMETERS(...)

The overall performance requirement is derived from the values specified for the PERFORMANCE and USAGE operands.

PERFORMANCE = *BY-PROGRAM / *BY-CATALOG / *STD / *HIGH / *VERY-HIGH / *USER-MAXIMUM

Specifies the performance attribute of the file. This governs the priority required for the I/O operations designated in the USAGE operand. The highest permissible performance attribute is defined in the user entry (see the output of the SHOW-USER-ATTRIBUTES command)

PERFORMANCE = *BY-PROGRAM

The value set in the program is used.

PERFORMANCE = *BY-CATALOG

The value specified in the catalog entry is used.

PERFORMANCE = *STD

There are no special performance attributes for the file, so a cache should not be used to process it.

PERFORMANCE = *HIGH

The file should be processed via a cache (high performance priority). Only users with the DMS tuning privilege CONCURRENT-USE or EXCLUSIVE-USE for the pubset can select this value (see the output of the SHOW-USER-ATTRIBUTES command, DMS-TUNING-RESOURCES output field).

PERFORMANCE = *VERY-HIGH

A cache should be used to process the file. The referenced data in the file should be kept in the cache permanently (highest performance priority, only available in the GS cache medium). The cached data is not cleared from the cache until the file is closed. Only users with the DMS tuning privilege EXCLUSIVE-USE for the pubset can select this value (see the output of the SHOW-USER-ATTRIBUTES command, *DMS-TUNING-RESOURCES* output field).

Note All the cache segments allocated to this file are locked until the file is closed. If there are many files open with this attribute at the same time, the cache memory available for normally cached files may as a result be so greatly restricted that data accesses to such files can no longer be implemented with an adequate level of performance.

PERFORMANCE = *USER-MAXIMUM

The file is assigned the highest performance attribute that is entered for the user in the user catalog.

USAGE = *BY-PROGRAM / *BY-CATALOG / *READ-WRITE / *WRITE / *READ
Specifies the I/O operations for which enhanced performance (caching) is required. The default is *READ-WRITE, i.e. the requirements apply to both read and write operations. If there is no special performance requirement for the file (PERFORMANCE=*STD) and if the cache area for the pubset is not defined to be available to all existing files (see CACHED-FILES=*ALL in the MODIFY-PUBSET-CACHE-ATTRIBUTES command), the USAGE operand has no effect on file processing.

USAGE = *BY-PROGRAM

The value specified in the program is used (see the FCB macro description in the "DMS Macros" manual [12]).

USAGE = *BY-CATALOG

The value specified in the catalog entry is used.

USAGE = *READ-WRITE

The enhanced performance requirement applies to both read and write operations.

USAGE = *WRITE

The enhanced performance requirement applies to write operations only.

USAGE = *READ

The enhanced performance requirement applies to read operations only.

Note With USAGE=*READ-WRITE or *WRITE, write caching is performed only if the conditions for the DISK-WRITE file attributes are met (see CREATE-FILE or MODIFY-FILE-ATTRIBUTES). For write caching with DISK-WRITE=*IMMEDIATE, the cache medium being used must be failsafe (see the *CACHE-MEDIUM* output field of the SHOW-MASTER-CATALOG-ENTRY command with INFORMATION=*USER).

ISAM-ATTRIBUTES = *BY-PROGRAM / *PARAMETERS(...)

Parameters relating to ISAM processing.

ISAM-ATTRIBUTES = *BY-PROGRAM

Information relating to ISAM-specific features is taken from the program or from the file specified in the DATA-ATTRIBUTES operand.

ISAM-ATTRIBUTES = *PARAMETERS(...)

Information relating to ISAM-specific features is derived from the values specified for the operands which follow.

KEY-POSITION = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / <integer 1..32767>

Specifies the position of the ISAM key within the record. In variable-length records allowance must be made for a 4-byte record length and control field. There is no fixed position for ISAM keys, but the key must always be in the same position throughout a file.

KEY-POSITION = *BY-DATA-ATTRIBUTES

The value of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, KEY-POSITION defaults to *BY-PROGRAM.

KEY-POSITION = *BY-PROGRAM

The value specified in the program is used (see the FCB macro description in the "DMS Macros" manual [12])

KEY-POSITION = *BY-CATALOG

The value set in the catalog is used.

KEY-POSITION = <integer 1..32767>

Specifies the position of the ISAM key within the record.

KEY-LENGTH = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / <integer 1..255>

Length of the ISAM key in bytes. Each record in the file must use the specified key length. The sum of the lengths specified in KEY-LENGTH, LOGICAL-FLAG-LENGTH and VALUE-FLAG-LENGTH must not come to more than 255 during processing.

KEY-LENGTH = *BY-DATA-ATTRIBUTES

The value of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, KEY-LENGTH defaults to *BY-PROGRAM.

KEY-LENGTH = *BY-PROGRAM

The value specified in the program is used.

KEY-LENGTH = *BY-CATALOG

The value set in the catalog is used.

KEY-LENGTH = <integer 1..255>

Specifies the length of the ISAM key in bytes.

VALUE-FLAG-LENGTH = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / <integer 0..255>

For K-ISAM files only: defines the length (in bytes) of the value flag in the ISAM index. This flag may form part of the ISAM record index together with the ISAM key and the logical flag. The sum of the lengths specified in KEY-LENGTH, LOGICAL-FLAG-LENGTH and VALUE-FLAG-LENGTH must not come to more than 255 during processing.

VALUE-FLAG-LENGTH = *BY-DATA-ATTRIBUTES

The value of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, VALUE-FLAG-LENGTH defaults to *BY-PROGRAM.

VALUE-FLAG-LENGTH = *BY-PROGRAM

The value set in the program is used.

VALUE-FLAG-LENGTH = *BY-CATALOG

The value set in the catalog is used.

VALUE-FLAG-LENGTH = <integer 0..255>

VALUE-FLAG-LENGTH = 0: the ISAM index does not include a value flag.

There is a difference in the treatment of value flags in NK-ISAM (BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK) and K-ISAM (BLOCK-CONTROL-INFO=*PAMKEY). In K-ISAM they are evaluated block by block and propagated to the next higher index entry, as specified in the PROPAGATE-VALUE-FLAG operand. In NK-ISAM no flags are evaluated for the index entry.

Note

If the “secondary key” function is used, the ISAM key must not include a value flag.

PROPAGATE-VALUE-FLAG = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *MINIMUM / *MAXIMUM / *BY-CATALOG

For K-ISAM files only (BLOCK-CONTROL-INFO=PAMKEY): defines how the value flag is to be propagated to the index entries. (NK-ISAM ignores PROPAGATE-VALUE-FLAG specifications)

PROPAGATE-VALUE-FLAG = *BY-DATA-ATTRIBUTES

The value of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, PROPAGATE-VALUE-FLAG defaults to *BY-PROGRAM.

PROPAGATE-VALUE-FLAG = *BY-PROGRAM

The value specified in the program is used (see the FCB macro description in the “DMS Macros” manual [12]).

PROPAGATE-VALUE-FLAG = *MINIMUM

The lowest value for the value flag within a data or index block is propagated to the next higher level.

PROPAGATE-VALUE-FLAG = *MAXIMUM

The highest value of the flag in the data/index block is propagated.

PROPAGATE-VALUE-FLAG = *BY-CATALOG

The value specified in the catalog entry is used.

LOGICAL-FLAG-LENGTH = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / <integer 0..255>

For K-ISAM files only: defines the length in bytes of a logical flag in the ISAM index; the maximum length is governed by the length of the ISAM key, in that the sum of the lengths specified in KEY-LENGTH, LOGICAL-FLAG-LENGTH and VALUE-FLAG-LENGTH must not come to more than 255 during processing.

LOGICAL-FLAG-LENGTH = *BY-DATA-ATTRIBUTES

The value of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, LOGICAL-FLAG-LENGTH defaults to *BY-PROGRAM.

LOGICAL-FLAG-LENGTH = *BY-PROGRAM

The value specified in the program is used (see the FCB macro description in the "DMS Macros" manual [12]).

LOGICAL-FLAG-LENGTH = *BY-CATALOG

The value specified in the catalog entry is used.

LOGICAL-FLAG-LENGTH = <integer 0..255>

LOGICAL-FLAG-LENGTH=0: the ISAM index does not include a logical flag. The ISAM key in the ISAM index may be followed by a logical flag in which selection criteria are defined bit by bit and encoded in binary. The ISAM key in the ISAM index may be followed by a logical flag in which selection criteria are defined bit by bit and encoded in binary. In K-ISAM files, all the logical flags of a block are evaluated and the result is propagated to the next higher index entry. There is no logical flag evaluation with NK-ISAM files.

Note

If the "secondary key" function is used, the ISAM key must not include a logical flag.

PADDING-FACTOR = *BY-PROGRAM / <integer 0..99>

Only for ISAM files created sequentially: the block padding factor specifies how much space to keep free in the data block for subsequent extension of the file (as a percentage of the block length defined in BUFFER-LENGTH). The PADDING-FACTOR specification thus affects the block splitting rate when a file is extended nonsequentially. K-ISAM requests a new logical block for the current record if the free space would otherwise fall below the padding factor. NK-ISAM does not request a new logical block until the free space has already fallen below the padding factor.

PADDING-FACTOR = *BY-PROGRAM

The value specified in the program is used.

PADDING-FACTOR = <integer 0..99>

A PADDING-FACTOR specification has different effects on NK-ISAM and K-ISAM. For NK-ISAM the block is filled *at least* up to the specified padding limit, for K-ISAM *at most* up to the padding limit

POOL-LINK = *BY-PROGRAM / <name 1..8>

Only for ISAM files processed in ISAM pools (NK-ISAM): specifies the pool link name to be recorded in the TFT. The file blocks required for processing are transferred from disk to the ISAM pool and buffered there.

POOL-LINK = *BY-PROGRAM

The pool link name specified in the program is used.

POOL-LINK = <name 1..8>

At OPEN time the pool link name is passed to NK-ISAM. This name must have been allocated using the ADD-ISAM-POOL-LINK command to an ISAM pool created with the CREATE-ISAM-POOL command.

POOL-SIZE = *BY-PROGRAM / <integer 128..1048576 2Kbyte>

For NK-ISAM files: Determines the size of the file-specific ISAM pool in units of 2048 bytes. The specification does not relate to the ISAM pool addressed with POOL-LINK.

WRITE-IMMEDIATE = *BY-PROGRAM / *NO / *YES

Controls how updated blocks are written out to disk. WRITE-IMMEDIATE=*YES applies implicitly to shared updating and to cross-task ISAM pools, which means that updated blocks are always written out to disk immediately.

WRITE-IMMEDIATE = *BY-PROGRAM

The value specified in the program is used.

WRITE-IMMEDIATE = *NO

An updated block is not written out to disk until the contents of the associated buffer area need to be replaced or the file is closed.

WRITE-IMMEDIATE = *YES

Each updated block is written out to disk immediately, thus ensuring consistency at all times between the data on disk and in virtual memory. The downside is a higher I/O frequency.

READ-IN-ADVANCE = *BY-PROGRAM / *YES / *NO

Specifies whether I/Os should be overlapped.

READ-IN-ADVANCE = *BY-PROGRAM

The setting defined in the program is used.

READ-IN-ADVANCE = *YES

If a second I/O area is defined in the program, read operations can be overlapped. With NK-ISAM, “overlapped processing” means that adjacent blocks are also read into the ISAM pool. READ-IN-ADVANCED =*YES should be used only for reads which are primarily sequential. With K-ISAM, “overlapped processing” means that if two I/O buffers are specified in the FCB a block can be read from one buffer to the other in the course of a read operation.

READ-IN-ADVANCE = *NO

Reads are not overlapped.

DUPLICATE-KEY = *BY-PROGRAM / *YES / *NO

Specifies whether the same ISAM key is allowed to occur more than once when records are being added using PUT or STORE macros

DUPLICATE-KEY = *BY-PROGRAM

The value defined in the program is used.

DUPLICATE-KEY = *YES

If a number of records have the same ISAM key, they do not overwrite each other but are written one after the other in the order of their creation. With NK-ISAM, records with identical keys have an 8-byte timestamp added to them internally. This must be allowed for when defining the record length.

See the section on overflow blocks in the “Introductory Guide to DMS” [13].

DUPLICATE-KEY = *NO

If the record being added has an ISAM key which has already occurred, the PUT macro causes the program to branch to the EXLST “DUPEKY” exit, while the STORE macro causes the existing record to be overwritten.

SUPPORT = *TAPE(...)

The file to be processed is a tape file.

VOLUME-LIST = *CATALOG(...) / *TEMPORARY (...) / *TAPE-SET (...)

Specifies the volume serial numbers (VSNs) of the tapes which are to be processed. With work files (DEVICE-TYPE= *WORK), the ADD-CATALOG-VOLUME operand is ignored and the TAPE-SET-NAME operand is rejected.

VOLUME-LIST = *CATALOG(...)

The VSNs from the volume table in the catalog entry are transferred to the volume list.

ADD-CATALOG-VOLUME = *NONE / *ANY (...) /**list-poss(255): <alphanum-name 1..6>**

*This operand is ignored for work files (DEVICE-TYPE=*WORK).*

As well as the VSNs from the volume table in the catalog entry, other VSNs can be added to the volume list. These VSNs are then also added to the volume table in the catalog entry.

ADD-CATALOG-VOLUME = *NONE

No other VSNs are added to the volume list.

ADD-CATALOG-VOLUME = *ANY (...)

The selection of VSNs is made either by the operating personnel (in response to a message on the console) or, where available, by the MAREN utility. This specification is ignored for the dummy file *DUMMY.

NUMBER-OF-DEVICES = 1 / <integer 1..9>

Number of extra VSNs to add to the volume list.

ADD-CATALOG-VOLUME = list-poss (255): <alphanum-name 1..6>

Extra VSNs to add to the volume list.

VOL-SEQUENCE-NUMBER = *NONE / *FROM-START-POSITION (...) / list-poss (255): <integer 1..255>

Defines which VSNs from the volume list are to be added to the volume table in the TFT entry and which order to add them in. This order has a significant impact on subsequent processing of the file.

VOL-SEQUENCE-NUMBER = *NONE

All the VSNs in the volume list are added to the volume table in the TFT entry.

VOL-SEQUENCE-NUMBER = *FROM-START-POSITION (...)

Indicates that VSNs from the volume table are to be added to the volume table in the TFT entry starting from a specified VSN.

START-POSITION = <integer 1..255>

All VSNs from the specified position onwards are added to the volume table in the TFT entry.

VOL-SEQUENCE-NUMBER = list-poss (255): <integer 1..255>

Selected VSNs in the volume list which are to be added to the volume table in the TFT entry. A number given here (such as 12) refers to the position of a VSN in the volume list (in this case the twelfth VSN). Thus the number entered here is the file section number.

VOLUME-LIST = *TEMPORARY(...)

The volume table of the existing catalog entry remains unchanged and is ignored while the file is being processed.

PROCESS-VOLUME = list-poss (255): <alphanum-name 1..6>

Defines a temporary list of VSNs for processing. This list forms the volume list and the volume table of the generated TFT entry.

VOLUME-LIST = *TAPE-SET(...)

*This operand is rejected for work files (DEVICE-TYPE=*WORK).*

Selects the tape file VSN set which was created or extended using CREATE- or EXTEND-TAPE-SET. The catalog entry must not yet have any VSNs in its volume table. The volume list is formed from the VSNs of a TST entry. All the VSNs in the volume list are added to the volume table of the catalog entry and to the volume table of the TFT entry.

TAPE-SET-NAME = <alphanum-name 1..4>

A maximum of 255 VSNs from the specified TST entry are added to the volume list, starting at the current VSN. The TFT entry which is generated is linked to the specified TST entry. If this TST entry does not exist, a TST entry is created with the specified name and an empty volume list. The final tape sequence for the (output) file is not fixed until OPEN processing is initiated. A reference to a tape set is allowed only for output files with standard labels.

FILE-SET-IDENTIFIER = *BY-TAPE-SET / <alphanum-name 1..6>

File set identifier of the TST entry. The VSN given here must match the file set identifier in the TST entry.

FILE-SET-IDENTIFIER = *BY-TAPE-SET

The file set identifier is defined by the TST entry.

FILE-SET-IDENTIFIER = <alphanum-name 1..6>

The VSN given here must match the file set identifier in the TST entry.

DEVICE-TYPE = *ANY / <device>

Device type for the required tapes. Only device and volume types known in the system are accepted. In interactive mode, DEVICE-TYPE=? calls up a list of the available device and volume types.

If a volume type of WORK is specified, a work tape is made available when the file is opened.

LABEL-PROCESSING = *BY-PROGRAM / *PARAMETERS(...)

Label processing options.

LABEL-PROCESSING = *BY-PROGRAM

The settings specified in the program are used.

LABEL-PROCESSING = *PARAMETERS(...)

The values set in the operands which follow are used.

LABEL-TYPE = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *NO / *NON-STD / *STD(...)

File label type. For existing tape files the label standard specified in the VOL1 label always applies. For output files (OPEN OUTIN/OUTPUT) the LABEL-TYPE operand is evaluated and the label standard in the VOL1 label is defined or modified accordingly.

LABEL-TYPE = *BY-DATA-ATTRIBUTES

The value of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, LABEL-TYPE defaults to *BY-PROGRAM.

LABEL-TYPE = *BY-PROGRAM(...)

The values defined in the program are used.

LABEL-TYPE = *NO

No file labels are read or written (no file label processing). If the tape has standard labels, the system processes the tape labels and checks for access authorization.

LABEL-TYPE = *NON-STD

The tape file has/is given nonstandard labels; file label processing is done in the user program. If the volume has standard labels, the system performs tape label processing and checks for access authorization.

LABEL-TYPE = *STD(...)

Selects standard labels for the file. The volumes containing the file must also have standard labels.

DIN-REVISION-NUMBER = *BY-PROGRAM / <integer 0..3>

The file and the volume have/are given standard labels conforming to the designated interchange level of DIN standard 66029.

- 0: only 2 HDR labels are generated (oldest standard label format; old BS2000 or BS1000 or IBM)
- 1: DIN standard 66029 dated August 1972
- 2: DIN standard 66029 dated June 1976
- 3: DIN standard 66029 dated May 1979

If the tape already contains files or file sections, the value specified here must be compatible with the label standard in the VOL1 label.

	LABEL=*STD(DIN-REVISION- NUMBER=)			
	0	1	2	3
DIN 66029 interchange level Edition	-	1 8/1972	2 6/1976	3 3/1978
Label standard in VOL1 label	(blank)↳	1	2	3
CODE=*ISO-7/ *OWN	not allowed	STD blocks converted to non- standard blocks REC-FORM=V: conversion to record format D REC-SIZE> 9999 or BUF-LEN> 9999 OPEN error	STD blocks converted to non- standard blocks REC-FORM=V: conversion to record format D REC-SIZE> 9999 or BUF-LEN> 9999 OPEN error	STD blocks converted to non- standard blocks REC-FORM=V: conversion to record format D REC-SIZE> 9999 or BUF-LEN> 9999 OPEN error
CODE=*EBCDIC			STD blocks converted to non-standard blocks	STD blocks converted to non- standard blocks
Access method			SAM only	SAM only
RECORD-FORMAT = *UNDEFINED				not allowed for out- put files; converted to DIN-REV-NUM=2

Table 23: Effect of the LABEL=*STD(DIN-REV-NUM=...) operand

Restrictions

- DIN-REVISION-NUMBER=1 is assumed for:
 - RECORD-FORMAT=*VARIABLE and CODE=*EBCDIC
 - BUFFER-LENGTH=*STD(...)
 - ACCESS-METHOD=*UPAM or ACCESS-METHOD=*BTAM
- If DIN-REVISION-NUMBER is 0, CODE must be *EBCDIC.
- If the number in the label standard indicator (VOL1 label) is lower than the number given in the DIN-REVISION-NUMBER operand, the label standard indicator number applies.
- In all other cases: DIN-REVISION-NUMBER=3 applies (e.g. with CODE=*ISO7 and RECORD-FORMAT=*VARIABLE).

For a file opened in INPUT, INOUT, EXTEND or REVERSE mode, the system ignores the number specified here and refers to the number (label standard indicator) given in the volume label (VOL1). For an output file, the significant number is the number given here or, if only *STD is specified, the most recent number; but note the following: If the allocated volume already contains one or more files or file sections, the number must match the label standard indicator in the first volume label. Otherwise the label standard indicator will be set according to the specified number.

This if a volume already contains one or more files or file sections and the label standard indicator in the first volume label is lower than the implicitly assumed standard version (number), the version number is taken from the volume label.

BYPASS-LABEL-CHECK = *BY-PROGRAM / *NO-POSITIONING / *ABSOLUTE-POSITIONING(...) / *FORWARD-POSITIONING(...) / *BACKWARD-POSITIONING(...)

Allows users with the appropriate authorization in their user entry (see the TPIGNORE output field of the SHOW-USER-ATTRIBUTES command) to bypass label checking so as to process tapes which were written on other operating systems or which have a structure and a label format that the system does not recognize. Code checking is also bypassed. If the tape is written in a code other than EBCDIC or ISO, the user must work with a dedicated code table (CODE= *OWN). A BYPASS-LABEL-CHECK specification applies only for the duration of file processing; it is not added to the catalog.

BYPASS-LABEL-CHECK = *BY-PROGRAM

The setting which applies in the program is used.

BYPASS-LABEL-CHECK = *NO-POSITIONING

Label handling is bypassed; header labels are not read or check; the tape position is not changed.

BYPASS-LABEL-CHECK = *ABSOLUTE-POSITIONING(...)

Absolute positioning. The tape is transported to the tape mark indicated in the next parameter. Counting from start of tape.

TAPE-MARK-NUMBER = <integer 0..32767>

Number of the tape mark to transport the tape to. TAPE-MARK-NUMBER = 0: position at start of tape.

BYPASS-LABEL-CHECK = *FORWARD-POSITIONING(...)

Forward positioning. The tape is wound forward by the number of tape marks indicated in the next parameter. Counting relative to the current tape position. No label checking is done.

NUMBER-OF-TAPE-MARKS = <integer 0..127>

Number of tape marks by which the tape is to be forwarded when the file is opened. NUMBER-OF-TAPE-MARKS = 0: the tape is not repositioned

BYPASS-LABEL-CHECK = *BACKWARD-POSITIONING(...)

Backward Positioning. The tape is wound back by the number of tape marks indicated in the next parameter. Counting relative to the current tape position. No label checking is done.

NUMBER-OF-TAPE-MARKS = <integer 0..127>

Number of tape marks by which the tape is to be reversed when the file is opened. NUMBER-OF-TAPE-MARKS=0: the tape is not repositioned

PROTECTION-LEVEL = *BY-PROGRAM / *LOW(...) / *HIGH(...)

Protection level (level of security) that label checking is to provide. Affects file processing in batch mode only. This operand is allowed only for tape or files with standard labels.

PROTECTION-LEVEL = *BY-PROGRAM

The setting selected in the program is used.

PROTECTION-LEVEL = *LOW(...)

If file processing is running in batch mode under the ID of the system administrator or the tape or file owner and appropriate authorization is allocated in the user entry (see the TPIGNORE output field of the SHOW-USER-ATTRIBUTES command), certain label processing error messages are suppressed.

OVERWRITE-PROTECTION = *NO / *YES

Indicates whether any other label checks are to be performed in addition to write protection.

OVERWRITE-PROTECTION = *NO

No other label checks are to be performed in addition to write protection.

OVERWRITE-PROTECTION = *YES

The system performs additional checks:

- If a new tape file is placed after an existing file on the tape, the labels of the preceding file are checked.
- the expiration date of the new file must not be later than that of the preceding file.
- ACCESS=READ must not be declared for the new file if ACCESS=WRITE has been declared for the preceding file.

PROTECTION-LEVEL = *HIGH(...)

Error messages are displayed on the console in batch mode. If the job is running under a user ID with TPIGNORE=YES authorization in the user catalog, operating personnel may ignore the error messages.

OVERWRITE-PROTECTION = *NO / *YES

See the PROTECTION-LEVEL=*LOW operand.

TAPE-MARK-WRITE = *BY-PROGRAM / *YES

Only for tape files without standard labels: specifies whether tape marks are written, i.e. the TAPE-MARK-WRITE operand is evaluated only when a tape file with LABEL set to *NO or *NON-STD is opened. Tape files with LABEL set to *STD (DIN-REV-NUM=n) are by default given tape marks after the labels.

TAPE-MARK-WRITE = *BY-PROGRAM

The value set in the program is used.

TAPE-MARK-WRITE = *YES

There is a tape mark after the label. Tape files without labels: the tape mark is placed at the start of the tape.

CODE = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / *EBCDIC / *ISO7 / *ISO7D / *OWN

For SAM or BTAM tape files: code in which the data and labels of the file are processed. Code translation is possible only for files with nonstandard blocks and may modify the record length field on output in locate mode. The value of the EBCDIC-TRANSLATION operand must be taken into account.

CODE = *BY-DATA-ATTRIBUTES

The value of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, CODE defaults to *BY-PROGRAM.

CODE = *BY-PROGRAM

The setting which applies in the program is used.

CODE = *BY-CATALOG

The setting defined in the catalog entry is used.

CODE = *EBCDIC

No code translation is required during processing.

Note

The German and international character sets use the same coding. Differences may arise due to the different key mapping on the display unit keyboard (TRANSDATA 8161). The same applies to CODE=*ISO7.

CODE = *ISO7

The tape file is written in ISO 7-bit code, which means that EBCDIC code is converted to ISO 7-bit on output and ISO 7-bit code is converted to EBCDIC on input. Code conversion is based on the international ISO table.

Notes

- The block length must be given in the form BUFFER-LENGTH=<integer 1..32767>, as otherwise the output blocks will be preceded by a (hexadecimal) PAM key. The same applies to CODE=*OWN.
- For output in locate mode with variable record format (RECORD FORMAT=*VARIABLE), the contents of the record length field (i.e. the first four bytes) are modified. In move mode the record length field is not modified. The same applies to CODE=*OWN.
- The German and international character sets use the same coding. Differences may arise due to the different key mapping on the display unit keyboard (TRANSDATA 8161). The same applies to CODE=*EBCDIC.

CODE = *ISO7D

Code conversion is as for CODE=*ISO7. It is, however based on the German ISO table rather than the international version.

CODE = *OWN

All tape I/O is performed on the basis of user-defined code translation tables. The addresses of these tables must have been defined in the file control block (FCB) of the program (using the TRTADR and TRTADW parameters in the FCB macro call).

Notes

- The block length must be given in the form BUFFER-LENGTH=<integer 1..32767>, as otherwise the output blocks will be preceded by a (hexadecimal) PAM key. The same applies to CODE=*ISO7.
- For output in locate mode with variable record format (RECORD FORMAT=*VARIABLE), the contents of the record length field (i.e. the first four bytes) are modified. In move mode the record length field is not modified. The same applies to CODE=*ISO7.

EBCDIC-TRANSLATION = *BY-PROGRAM / *YES / *NO

Only for tape files used as input files and not created using CODE= *EBCDIC: defines how the code in the file is to be translated when the file is read.

EBCDIC-TRANSLATION = *BY-PROGRAM

The setting selected in the program is adopted.

EBCDIC-TRANSLATION = *YES

ISO 7-bit code or OWN code is translated to EBCDIC.

EBCDIC-TRANSLATION = *NO

No translation to EBCDIC. ISO 7-bit code is translated to an 8-bit format with a leading zero.

FILE-SEQUENCE = *BY-PROGRAM / *BY-CATALOG / *UNKNOWN / *NEW / <integer 0..9999>

Specifies the sequence number of a tape file within a file set. If there are several files of the same name on a tape, access is under FILE-SEQUENCE control. The same applies to MF/MV sets. If the VOLUME-LIST *TAPE-SET(...) operand is also set, only FILE-SEQUENCE=*NEW or FILE-SEQUENCE=1 is allowed.

FILE-SEQUENCE = *BY-PROGRAM

The value set in the program is used.

FILE-SEQUENCE = *BY-CATALOG

If there is a file sequence number in the catalog entry, it is transferred to the TFT entry. If not, no file sequence number is added to the catalog entry, and *BY-CATALOG is added to the TFT entry.

FILE-SEQUENCE = *UNKNOWN

If there is a file sequence number in the catalog entry, it is transferred to the TFT entry. If not, no file sequence number is added to the catalog entry, and *UNKNOWN is added to the TFT entry. The effect of this on a foreign tape file with standard labels is that, when an attempt is made to open the file, the tape is searched for the file and positioned accordingly.

FILE-SEQUENCE = *NEW

Allowed only for files which have been opened (OPEN OUTPUT) but not yet actually created. This operand adds a new file to an existing file set. The tape is transported to the end of the file set, and the new file is written after the last existing file in the set. The system reports an OPEN error if it fails to find the end of the file set. The sequence number of the new file is the sequence number of the old last file plus 1. If there are no file on the tape at the time the file is opened, the file is made the first in the file set and accordingly assigned a file sequence number of 1.

FILE-SEQUENCE = <integer 0..9999>

Specifies the sequence number of a file with a file set; FILE-SEQUENCE=0, like FILE-SEQUENCE=1, designates the first file in the set. If the file has already been cataloged with a creation date, the file sequence number that the user specifies must match the number which appears in the catalog entry.

If a new file is to be created, it is appended to the end of the file set, which means that the file sequence number must be 1 higher than the last existing file in the file set.

Note

If the sequence number of an existing file in the file set is specified, the existing file will be overwritten.

CHECKPOINT-WRITE = *BY-PROGRAM / *PARAMETERS(...)

Controls whether and when a checkpoint is automatically written at end of volume.

CHECKPOINT-WRITE = *BY-PROGRAM

The setting selected in the program is adopted.

CHECKPOINT-WRITE = *PARAMETERS(...)**CHKPT-AT-BLOCK-LIMIT = *BY-PROGRAM**

The setting selected in the program is adopted.

CHKPT-AT-BLOCK-LIMIT = *YES

A checkpoint is written automatically when the maximum number of logical blocks per tape as defined in the BLOCK-LIMIT operand is reached.

CHKPT-AT-FORCED-EOV = *BY-PROGRAM

The setting selected in the program is adopted.

CHKPT-AT-FORCED-EOV = *YES

Checkpoint at forced end of volume. A checkpoint is written automatically in response to each FEOV macro call in the Assembler program.

RESTART-USAGE = *BY-PROGRAM / *DUMMY

Indicates whether, in the event of a restart (RESTART-PROGRAM command), the file specified in the FILE-NAME operand is to continue being processed or is to be treated as a DUMMY file.

RESTART-USAGE = *BY-PROGRAM

The setting selected in the program is adopted.

RESTART-USAGE = *DUMMY

In the event of a restart the file is treated as a DUMMY file.

BLOCK-LIMIT = *BY-PROGRAM / <integer 1..999999>

Only when creating SAM tape files with standard labels: maximum number of logical blocks of the file per tape.

BLOCK-LIMIT = *BY-PROGRAM

The setting selected in the program is adopted.

BLOCK-LIMIT = <integer 1..999999>

Specifies how many logical data block are allowed to be written on one tape. Once this limit is reached, tape swapping is initiated (EOV processing). The user is sent an error message if the end of the tape is reached before the number of blocks specified by BLOCK-LIMIT has been written.

BLOCK-OFFSET = *BY-DATA-ATTRIBUTES / *BY-PROGRAM / *BY-CATALOG / *BY-HDR2 / <integer 0..99 byte>

Only for SAM tape files with BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK or SAM tape files with nonstandard blocks: defines the buffer offset, i.e. the length in bytes of a field inserted at the start of each logical data block. If present, this field contains the block control information/block length field.

BLOCK-OFFSET = *BY-DATA-ATTRIBUTES

The value of the file specified in the DATA-ATTRIBUTES operand is used. If DATA-ATTRIBUTES is *STD, BLOCK-OFFSET defaults to *BY-PROGRAM.

BLOCK-OFFSET = *BY-PROGRAM

The value set in the program is used.

If there is no value set in the program, the following defaults apply:

- tape file with BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK:
 - BLOCK-OFFSET=16 applies with ACCESS-METHOD=*SAM
 - BLOCK-OFFSET=12 applies with ACCESS-METHOD=*UPAM
- tape file with BLOCK-CONTROL-INFO=*NO:
 - BLOCK-OFFSET=4 applies with RECORD-FORMAT=*VARIABLE
 - BLOCK-OFFSET=0 applies with RECORD-FORMAT=*FIXED/*UNDEFINED

BLOCK-OFFSET = *BY-CATALOG

The value set in the catalog is used.

BLOCK-OFFSET = *BY-HDR2

The BLOCK-OFFSET value is taken from the HDR2 label. If there is no HDR2 label, or if the "buffer offset" field in the HDR2 label contains blanks (X'4040'), the default is used (see *BY-PROGRAM).

BLOCK-OFFSET = <integer 0..99 byte>

Specifies the length of the buffer offset.

For variable-length records (RECORD-FORMAT=*VARIABLE) the buffer offset may have a length between 0 and 4 bytes. If BLOCK-OFFSET=4 is set, this field contains the current block length.

TAPE-WRITE = *BY-PROGRAM / *DEVICE-BUFFER / *IMMEDIATE

Only for files on tape cartridges (CREATE-FILE/MODIFY-FILE-ATTRIBUTES command, DEVICE-TYPE operand): defines whether output is to be buffered.

TAPE-WRITE = *BY-PROGRAM

The value specified in the program is used.

TAPE-WRITE = *DEVICE-BUFFER

Output is buffered by the device controller, resulting in a high data transfer rate.

TAPE-WRITE = *IMMEDIATE

Output is not buffered.

DESTROY-OLD-CONTENTS = *BY-CATALOG / *NO / *YES

The user can specify whether other files on the tape are to be destroyed by overwriting after EOF/EOV processing. The operand has the same function as the DESTROY parameter in the catalog entry, but it takes precedence over the catalog entry created by the CREATE-FILE or MODIFY-FILE-ATTRIBUTES command, DESTROY-BY-DELETE operand. Values specified for this operand are not transferred to the FCB or the catalog entry.

DESTROY-OLD-CONTENTS = *BY-CATALOG

When the file is processed, the value assigned to DESTROY-BY-DELETE in the catalog entry takes effect.

DESTROY-OLD-CONTENTS = *NO

The old contents are not destroyed.

DESTROY-OLD-CONTENTS = *YES

The data on the remainder of the tape is destroyed once the EOF/EOV labels have been written.

IO-CHAINING = *BY-PROGRAM / <integer 1..16>

Only for BTAM files and chained I/O: defines the number of logical blocks per transport unit (chaining factor specifying the of the I/O transport unit/transmission unit)

IO-CHAINING = *BY-PROGRAM

The value specified in the program is used.

IO-CHAINING = <integer 1..16>

Designates a number of logical blocks such that the length of the transport unit is computed as (IO-CHAINING × BUFFER-LENGTH). Although specifications in the program (BTAM macro, LEN operand) take precedence over the product (IO-CHAINING × BUFFER-LENGTH), IO-CHAINING must be specified in the ADD-FILE-LINK command if chained I/O is used.

STREAMING-MODE = *BY-PROGRAM / *YES

For BTAM files only: allows tapes to be processed in streaming mode. For better performance, make sure that only one streaming device is connected to a channel.

FILE-CLOSE-MSG = *STD / *NO / *YES

For SAM files the user can specify whether a message is to be written to SYSOUT on completion of CLOSE processing.

FILE-CLOSE-MSG = *STD

Default for tape disk files:

FILE-CLOSE-MSG = *NO

Default for tape files:

FILE-CLOSE-MSG = *YES

FILE-CLOSE-MSG = *NO

The file close message is suppressed.

FILE-CLOSE-MSG = *YES

The file close message is sent.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command successfully executed
	1	CMD0202	Syntactical or semantic error
	1	DMS0576	Invalid operand combination
	32	DMS0584	A state that does not allow the function to continue was reported during processing
	32	DMS05C7	Unexpected internal error in DMS
	64	CMD0216	User does not have the necessary privilege
	64	DMS0501	Requested catalog not available
	64	DMS0512	Requested catalog not available
	64	DMS051B	Requested user ID not in pubset Guaranteed messages: DMS051B, DMS0681
	64	DMS051C	User not authorized to access pubset Guaranteed messages: DMS051C, DMS0681
	64	DMS0535	Specified file not shareable
	64	DMS0585	Error detected during catalog processing or multihost processing
	64	DMS0586	Currently not possible to access or reserve volume
	64	DMS0587	Use of specified command restricted by system administrator
	64	DMS05FC	Specified user ID not in HOME pubset
	64	DMS060D	Invalid reference file name
	64	DMS0616	Volume set not accessible in SM pubset
	64	DMS0684	File does not exist Guaranteed message: DMS0684
	64	DMS0685	File not yet allocated storage space
	64	DMS06FF	BCAM connection severed
	130	DMS0524	System address space full
	130	DMS053C	No space in pubset catalog file
	130	DMS0582	File is currently locked or in use and cannot be processed
	130	DMS0585	Error detected during catalog processing or multihost processing
	130	DMS0586	Currently not possible to access or reserve volume
	130	DMS0594	Not enough virtual memory available
	130	DMS05C8	CE limit reached

Examples

Example 1: Creating a TFT entry with values from the catalog entry

```

/show-file-attr savlst.hsmsmac,inf=(org=yes)
%0000000027 :20SG:$USER1.SAVLST.HSMSMAC
%----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE) = READ-WRITE   IO(PERF) = STD           DISK-WRITE = IMMEDIATE
% REC-FORM = (V,N)         REC-SIZE = 0
% AVAIL = *STD
% WORK-FILE = *NO         F-PREFORM = *K           SO-MIGR = *ALLOWED
%:20SG: PUBLIC: 1 FILE RES= 27 FRE= 4 REL= 3 PAGES
/add-file-link link=edtsam,file-name=avlst.hsmsmac,
access-method=*by-cat,
rec-form=*by-cat,buffer-length=*by-cat,block-contr-info=*by-cat
/show-file-link link=edtsam,inf=*all
%-- LINK-NAME ----- FILE-NAME -----
% EDTSAM :20SG:$USER1.SAVLST.HSMSMAC
%----- STATUS -----
% STATE = INACTIVE ORIGIN = FILE
%----- PROTECTION -----
% RET-PER = *BY-PROG PROT-LEV = *BY-PROG
% BYPASS = *BY-PROG DESTROY = *BY-CAT
%----- FILE-CONTROL-BLOCK - GENERAL ATTRIBUTES -----
% ACC-METH = *BY-CAT OPEN-MODE = *BY-PROG REC-FORM = *BY-CAT
% REC-SIZE = *BY-PROG BUF-LEN = *BY-CAT BLK-CONTR = *BY-CAT
% F-CL-MSG = STD CLOSE-MODE = *BY-PROG
%----- FILE-CONTROL-BLOCK - DISK FILE ATTRIBUTES -----
% SHARED-UPD = *BY-PROG WR-CHECK = *BY-PROG IO(PERF) = *BY-PROG
% IO(USAGE) = *BY-PROG LOCK-ENV = *BY-PROG
%----- FILE-CONTROL-BLOCK - TAPE FILE ATTRIBUTES -----
% LABEL = *BY-PROG (DIN-R-NUM = *BY-PROG, TAPE-MARK = *BY-PROG)
% CODE = *BY-PROG EBCDIC-TR = *BY-PROG F-SEQ = *BY-PROG
% CP-AT-BLIM = *BY-PROG CP-AT-FEOV = *BY-PROG BLOCK-LIM = *BY-PROG
% REST-USAGE = *BY-PROG BLOCK-OFF = *BY-PROG TAPE-WRITE = *BY-PROG
% STREAM = *BY-PROG
%----- FILE-CONTROL-BLOCK - ISAM FILE ATTRIBUTES -----
% KEY-POS = *BY-PROG KEY-LEN = *BY-PROG POOL-LINK = *BY-PROG
% LOGIC-FLAG = *BY-PROG VAL-FLAG = *BY-PROG PROPA-VAL = *BY-PROG
% DUP-KEY = *BY-PROG PAD-FACT = *BY-PROG READ-I-ADV = *BY-PROG
% WR-IMMED = *BY-PROG POOL-SIZE = *BY-PROG
%----- VOLUME -----
% DEV-TYPE = *NONE T-SET-NAME = *NONE
% VSN/DEV = GVS2.2/D3435

```

All the file attributes for which *BY-CATALOG is specified in the ADD-FILE-LINK command are marked *BY-CAT in the TFT entry. When the file named *LST.HSMSRECALL* is accessed via its link name of *EDTSAM*, these attributes will be taken from the file's catalog entry.

Example 2: File attributes in the command

```

/create-file max.file.1
/show-file-attr max.file.1,org=*yes
%0000000003 :20SG:$USER1.MAX.FILE.1
% ----- ORGANIZATION -----
% FILE-STRUC = NONE          BUF-LEN      = NONE          BLK-CONTR = NONE
% IO(USAGE)  = READ-WRITE   IO(PERF)   = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = NONE          REC-SIZE   = 0
% AVAIL      = *STD
% WORK-FILE  = *NO          F-PREFORM = *K          SO-MIGR   = *ALLOWED
%:20SG: PUBLIC:          1 FILE RES=          3 FRE=          3 REL=          3 PAGES
/add-file-link link=output1,file-name=max.file.1,acc-method=*isam,
support=*disk(isam-attr=*par(key-pos=5,key-length=10))
/show-file-link link=output1,inf=*par(file=*yes)
%-- LINK-NAME ----- FILE-NAME -----
% OUTPUT1 :20SG:$USER1.MAX.FILE.1
% ----- FILE-CONTROL-BLOCK - GENERAL ATTRIBUTES -----
% ACC-METH = ISAM          OPEN-MODE = *BY-PROG   REC-FORM = *BY-PROG
% REC-SIZE = *BY-PROG     BUF-LEN   = *BY-PROG   BLK-CONTR = *BY-PROG
% F-CL-MSG = STD          CLOSE-MODE = *BY-PROG
% ----- FILE-CONTROL-BLOCK - DISK FILE ATTRIBUTES -----
% SHARED-UPD = *BY-PROG   WR-CHECK  = *BY-PROG   IO(PERF) = *BY-PROG
% IO(USAGE)  = *BY-PROG   LOCK-ENV  = *BY-PROG
% ----- FILE-CONTROL-BLOCK - TAPE FILE ATTRIBUTES -----
% LABEL      = *BY-PROG   (DIN-R-NUM = *BY-PROG, TAPE-MARK = *BY-PROG)
% CODE       = *BY-PROG   EBCDIC-TR = *BY-PROG   F-SEQ     = *BY-PROG
% CP-AT-BLIM = *BY-PROG   CP-AT-FEOV = *BY-PROG   BLOCK-LIM = *BY-PROG
% REST-USAGE = *BY-PROG   BLOCK-OFF  = *BY-PROG   TAPE-WRITE = *BY-PROG
% STREAM     = *BY-PROG
% ----- FILE-CONTROL-BLOCK - ISAM FILE ATTRIBUTES -----
% KEY-POS    = 5          KEY-LEN    = 10          POOL-LINK = *BY-PROG
% LOGIC-FLAG = *BY-PROG   VAL-FLAG   = *BY-PROG   PROPA-VAL = *BY-PROG
% DUP-KEY    = *BY-PROG   PAD-FACT   = *BY-PROG   READ-I-ADV = *BY-PROG
% WR-IMMED   = *BY-PROG   POOL-SIZE  = *BY-PROG

```

A new file, *MAX.FILE.1*, is created with the CREATE-FILE command. The attributes specified in the ADD-FILE-LINK command are transferred to the TFT entry. When the file is accessed via its link name, *OUTPUT1*, these attributes will be transferred to the file control block.

Example 3: Importing file attributes from a reference file

```

/cre-file abk.neu
/show-file-attr abk.isam,inf=(org=*yes)
%0000000126 :20SG:$USER1.ABK.ISAM
% ----- ORGANIZATION -----
% FILE-STRUC = ISAM          BUF-LEN      = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE   IO(PERF)   = STD            DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)        REC-SIZE   = 0
% KEY-LEN    = 30          KEY-POS    = 8
% AVAIL      = *STD
% WORK-FILE  = *NO         F-PREFORM  = *K            SO-MIGR   = *ALLOWED
%:20SG: PUBLIC:          1 FILE RES=          126 FRE=          5 REL=          3 PAGES
/add-file-link link=output2,file-name=abk.neu,
               data-attr=*from-file(abk.isam)
/show-file-link link=output2,inf=*all
%-- LINK-NAME ----- FILE-NAME -----
%  OUTPUT2          :20SG:$USER1.ABK.NEU
% ----- STATUS -----
% STATE            = INACTIVE          ORIGIN          = FILE
% ----- PROTECTION -----
% RET-PER          = *BY-PROG          PROT-LEV        = *BY-PROG
% BYPASS           = *BY-PROG          DESTROY         = *BY-CAT
% ----- FILE-CONTROL-BLOCK - GENERAL ATTRIBUTES -----
% ACC-METH         = ISAM              OPEN-MODE       = *BY-PROG          REC-FORM      = VARIABLE
% REC-SIZE         = 0                 BUF-LEN         = (STD,1)          BLK-CONTR    = PAMKEY
% F-CL-MSG         = STD               CLOSE-MODE      = *BY-PROG
% ----- FILE-CONTROL-BLOCK - DISK FILE ATTRIBUTES -----
% SHARED-UPD      = *BY-PROG          WR-CHECK        = *BY-PROG          IO(PERF)     = *BY-PROG
% IO(USAGE)       = *BY-PROG          LOCK-ENV        = *BY-PROG
% ----- FILE-CONTROL-BLOCK - TAPE FILE ATTRIBUTES -----
% LABEL           = *BY-PROG          (DIN-R-NUM     = *BY-PROG,          TAPE-MARK    = *BY-PROG)
% CODE            = *BY-PROG          EBCDIC-TR      = *BY-PROG          F-SEQ        = *BY-PROG
% CP-AT-BLIM      = *BY-PROG          CP-AT-FEOV     = *BY-PROG          BLOCK-LIM    = *BY-PROG
% REST-USAGE      = *BY-PROG          BLOCK-OFF      = 0                 TAPE-WRITE   = *BY-PROG
% STREAM          = *BY-PROG
% ----- FILE-CONTROL-BLOCK - ISAM FILE ATTRIBUTES -----
% KEY-POS         = 8                 KEY-LEN         = 30          POOL-LINK    = *BY-PROG
% LOGIC-FLAG      = 0                 VAL-FLAG        = 0           PROPA-VAL    = MINIMUM
% DUP-KEY         = *BY-PROG          PAD-FACT        = *BY-PROG          READ-I-ADV   = *BY-PROG
% WR-IMMED        = *BY-PROG          POOL-SIZE       = *BY-PROG
% ----- VOLUME -----
% DEV-TYPE        = *NONE              T-SET-NAME     = *NONE
% VSN/DEV         = GVS2.3/D3435

```

A new file, *ABK.NEU*, is created with the CREATE-FILE command. For processing purposes the new file is to have the same attributes as an existing file, *ABK.ISAM*. For this purpose *ABK.ISAM* is specified as the reference file in the ADD-FILE-LINK command (see DATA-ATTRIBUTES operand). The file attributes of the reference file specified in the ADD-FILE-LINK command are copied into the TFT entry. When the new file is processed by way of the link name *OUTPUT2*, these attributes are copied into the file control block.

ADD-IO-UNIT

Add input/output unit to configuration

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Device management
Domain:	DEVICE
Privileges:	OPERATING
Routing code:	G

Function

Systems support can dynamically extend the input/output configuration with the ADD-IO-UNIT command. New input/output units (devices, controllers, type S or type F channels) can be defined. New input/output units have to be specified in the order channel controller device. Up to 256 devices can be added to controller using the ADD-IO-UNIT command.

→ →

Dynamic I/O configuration change is supported for the following device classes:

- disk and tape devices
- printer
- data communication devices
- non-standard devices

The command is only accepted if the dynamic I/O configuration change was started successfully (see START-CONFIGURATION-UPDATE command).

Restrictions and peculiarities

The following procedures have to be completed before a new tape device that has been added using ADD-IO-UNIT can take up operation:

1. A depot has to be assigned to the device using the ADD-DEVICE-DEPOT command.
2. If MAREN is being used and the new device is to be subject to free tape allocation, the MARENUCP subsystem has to be stopped and restarted.
3. If the new device is being operated by a robot and if it is not contained in the ROBAR configuration file, the ROBAR-SV subsystem has to be stopped and restarted.

A new printer has to be added to the SPOOL parameter file before or after being added (SPSERVE statement ADD-SPOOL-DEVICE).

A new data communications device as to be made known in the DCM with the BCIN command before or after being added.

The IOTRACE subsystem only considers new input/output units, if it is restarted after changes have been made to the configuration.

Format

ADD-IO-UNIT
<p>UNIT = *CHANNEL(...) / *CONTROLLER(...) / *DEVICE(...)</p> <p>*CHANNEL(...)</p> <ul style="list-style-type: none"> TYPE = *<u>IBS</u>(...) / *<u>IBF</u>(...) *IBS(...) <ul style="list-style-type: none"> MODE = *<u>CNC</u> / *<u>CTC</u> / *<u>CVC</u> *IBF(...) <ul style="list-style-type: none"> MODE = *<u>FCP</u> <p>,CHANNEL-PATH-ID = <x-text 2..3></p> <p>*CONTROLLER(...)</p> <ul style="list-style-type: none"> NAME = <alphanum-name 2..2> / <x-text 4..4> ,TYPE = <alphanum-name 3..6> ,LOGICAL-ADDRESS = <u>0</u> / <integer 0..255> ,PATH = list-poss(8): *CHANNEL(...) *CHANNEL(...) <ul style="list-style-type: none"> CHANNEL-PATH-ID = <x-text 2..3> ,CONTROLLER-ADDRESS = <u>00</u> / <x-text 1..2> ,PATH-STATE = *<u>INCLUDED</u> / *<u>REMOVED</u> ,PORT-ID = *<u>NONE</u> / <x-text 2..16> ,DEVICE-RANGE = list-poss(8): *DEVICE(...) *DEVICE(...) <ul style="list-style-type: none"> FIRST-ADDRESS = <u>00</u> / <x-text 2..2> ,NUMBER-OF-DEVICES = <u>256</u> / <integer 1..256>

(Part 1 of 2)

```

*DEVICE(...)
  NAME = <alphanum-name 2..2> / <x-text 4..4> / *RANGE(...)
    *RANGE(...)
      | FIRST-NAME = <alphanum-name 2..2> / <x-text 4..4>
      | ,NUMBER-OF-DEVICES = <integer 2..256>
    ,TYPE = <x-text 2..4 without-odd>
    ,DEVICE-ADDRESS = <x-text 2..2>
    ,PATH = list-poss(8): *CHANNEL(...) / *CONTROLLER(...)
      *CHANNEL(...)
        | CHANNEL-PATH-ID = <x-text 2..3>
        | ,PATH-STATE = *INCLUDED / *REMOVED
      *CONTROLLER(...)
        | NAME = <alphanum-name 2..2> / <x-text 4..4>
        | ,PATH-STATE = *INCLUDED / *REMOVED
      ,PREFERRED-PATH = *NO / *YES(...)
        *YES(...)
          | CHANNEL-PATH-ID = <x-text 2..3>
        ,PAV-ALIAS-DEVICE = *NO / *YES(...)
          *YES(...)
            | DEVICE-ADDRESS = <x-text 2..2>
      ,STATE = *ATTACHED / *DETACHED

```

(Part 2 of 2)

Operands

UNIT = *CHANNEL(...) / ***CONTROLLER(...)** / ***DEVICE(...)**

Specifies which input/output unit is to be added to the configuration.

UNIT = *CHANNEL(...)

Specifies the channel to be added to the configuration.

TYPE = *IBS(...)

A type S bit-serial channel (IBS channel) is to be added.

MODE = *CNC / ***CTC** / ***CVC**

Specifies the mode of operation of the IBS channel.

MODE = *CNC

The channel is operated in “connection channel” mode (for the connection of controllers compatible with type S). This is the default value.

MODE = *CTC

The channel is operated in “channel-to-channel” mode.

MODE = *CVC

The channel is operated in “S/P converter” mode (for the connection of controllers compatible with type 2 to type S channels).

TYPE = *IBF(...)

A serial type FC channel is to be added.

MODE = *FCP

Specifies the mode in which the IBF channel is to operate:

The channel is operated in “fibre channel” mode (for connecting controllers with FC capability). This is the default value.

CHANNEL-PATH-ID = <x-text 2..3>

The number of the channel.

UNIT = *CONTROLLER(...)

The controller that is to be added to the configuration.

NAME = <alphanum-name 2..2> / <x-text 4..4>

Mnemonic device code of the controller: either two alphanumeric characters (A...Z, 0...9) or four hexadecimal digits in the range 1000 through FFFF.

TYPE = <alphanum-name 3..6>

Controller type. See [table 24 on page 2-96](#) for possible values.

LOGICAL-ADDRESS = 0 / <integer 0..255>

Specifies a logical address:

- from 0 to 15 for the controller on the type S,
- or 0 to 255 for the controller on the type FC channel (MODE=*FCP).

The default is logical address 0.

For the controller on the fibre channel this specifies the more significant byte of the LUNs (Logical Unit Numbers) of the devices connected to this controller. The low order byte of the LUN is defined for the device by the DEVICE-ADDRESS operand.

PATH = list-poss(8): *CHANNEL(...)

As many as eight input/output paths via which the controller can be accessed can be entered in a list.

CHANNEL-PATH-ID = <x-text 2..3>

Specifies the number of the channel to which the controller is linked.

CONTROLLER-ADDRESS = 00 / <x-text 1..2>

Specifies the physical controller address.

The default is address 00.

PATH-STATE = *INCLUDED / *REMOVED

Specifies whether the link between the channel and the controller is available.

The default is *INCLUDED, i.e. the connection is available.

PORT-ID = *NONE / <x-text 2..16>

Specifies whether the controller is connected to a Fibre Channel.

The default is *NONE, i.e. the controller is not connected to a Fibre Channel.

PORT-ID = <x-text 2..16>

When a controller is connected to a Fibre Channel the WWPN (World Wide Port Name) of the controller port (16 hexadecimal characters) must be specified. The /SHOW-DEVICE-CONFIGURATION INF=*INNER command for the controller can then be used to output the WWPN.

DEVICE-RANGE = list-poss(8): *DEVICE(...)

Lists up to eight address areas through which devices linked to the controller can be accessed. Each address area is specified by the first device address and the total number of device addresses that follow it. To determine a device address in the address area, 1 is added for each device, starting from the first address. The value of the final address must not exceed FFFF.

FIRST-ADDRESS = 00 / <x-text 2..2>

First device address.

NUMBER-OF-DEVICES = 256 / <integer 1..256>

Number of devices in the address area.

UNIT = *DEVICE(...)

A device that is to be added to the configuration.

NAME = <alphanum-name 2..2> / <x-text 4..4> / *RANGE(...)

Mnemonic device code of the device: either two alphanumeric characters (A...Z, 0...9) or four hexadecimal digits in the range 1000 through FFFF. A two-character alphanumeric mnemonic can always be selected for devices. A four-digit hexadecimal device mnemonic (1000 - FFFF) is permitted for all disk and tape devices and for devices of the type 6D (HNC).

A group of devices can be added with the aid of the *RANGE operand.

NAME = *RANGE(...)

A range of devices is added. Based on the mnemonics of the first device (FIRST-NAME operand), the mnemonics of the following devices are determined by adding 1 until the maximum number of devices defined in the NUMBER-OF-DEVICES operand is reached. The group can comprise as many as 256 devices. The maximum number has to be chosen so as not to exceed the value FFFF when the mnemonics are calculated.

FIRST-NAME = <alphanum-name 2..2> / <x-text 4..4>

Mnemonic device code of the first device.

NUMBER-OF-DEVICES = <integer 2..256>

Number of devices to be added.

TYPE = <x-text 2..4 without-odd>

Device type code (see "System installation" [46]).

DEVICE-ADDRESS = <x-text 2..2>

Device address.

For a device on the Fibre Channel this specifies the low order byte of the LUN (Logical Unit Number). The more significant byte of the LUN is defined on the controller using the LOGICAL-ADDRESS operand.

PATH = list-poss(8): *CHANNEL(...)/ *CONTROLLER(...)

Up to eight input/output paths can be specified in a list, via which a device or a group of devices can be reached. The entries can refer either on the channel or on the controller. A mixed entry of both is not possible.

PATH = list-poss(8): *CHANNEL(...)

As many as eight channels via which the device can be addressed can be specified in a list.

Channel-specific specifications are not currently supported on S servers and SUs /390. Devices can only be added via a controller (see PATH= *CONTROLLER).

CHANNEL-PATH-ID = <x-text 2..3>

Number of the channel to which the device is linked.

PATH-STATE = *INCLUDED / *REMOVED

Specifies whether the link between the channel and the device is available or not. The default is *INCLUDED, i.e. the connection is available.

PATH = list-poss(8): *CONTROLLER(...)

As many as eight controllers via which the device can be accessed can be specified in a list.

NAME = <alphanum-name 2..2> / <x-text 4..4>

The device code of the controller to which the device is linked.

PATH-STATE = *INCLUDED / *REMOVED

Specifies whether the link between the controller and the device is available or not. The default is *INCLUDED, i.e. the connection is available.

PREFERRED-PATH = *NO / *YES(...)

Specifies whether a path is to preferred on input/output if the device can be accessed via different paths. The default setting is *NO, i.e. there is no preferred path.

PREFERRED-PATH = *YES(...)

A specific input/output path is to be used with preference.

CHANNEL-PATH-ID = <x-text 2..3>

Designates the preferred path (Channel Path Identifier).

PAV-ALIAS-DEVICE = *NO / *YES(...)

Specifies whether the device is to be defined as an alias device of a PAV volume on the Fibre Channel. The default is *NO, i.e. no alias device is defined.

PAV-ALIAS-DEVICE = *YES(...)

The device is to be defined as an alias device of a PAV volume. The associated PAV base device must already be defined and it may not have a lower device number than the alias device (an assignment table with the mnemonic and device number is contained in the "System Installation" manual [46]).

DEVICE-ADDRESS = <x-text 2..2>

Alias address of the alias device. The address must be unique in the controller. For an alias device on the Fibre Channel the alias address must differ from the low order byte of the LUN.

Information on generating base and alias controllers is provided in the "System Installation" manual [46].

STATE = *ATTACHED / *DETACHED

Specifies the status (available or not available) in which the input/output unit is added to the configuration. The default is *ATTACHED, i.e. the new input/output unit is available. This status can be changed using the ATTACH-DEVICE or DETACH-DEVICE commands.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without error
	1	NKR0006	Syntax error
	64	CMD0216	Insufficient authorization
1	64	NKR0169	system error
2	64	NKR0169	Wrong parameter
3	64	NKR0169	Dynamic I/O configuration change was rejected
4	64	NKR0169	Dynamic I/O configuration change not supported
6	64	NKR0169	Internal input/output unit not defined
7	64	NKR0169	Input/output unit already defined
8	64	NKR0169	Input/output unit type not supported
10	64	NKR0169	No free table entry for input/output unit
12	64	NKR0169	Dynamic I/O configuration change was not started
14	64	NKR0169	Configuration changes not made in guest system
16	64	NKR0169	Another guest system currently being started
17	64	NKR0169	Another guest system currently being stopped
18	64	NKR0169	Dynamic I/O configuration change is not supported by one or several guest systems

Specification of controller type (UNIT=*CONTROLLER)

The TYPE operand characterized the control features in IORSF. The possible values are shown in the following table:

TYPE	DEV-TYP	CU	TA	SU	CP	DR	SV
CON001	0X	2	N	Y	I	N	Y
PRN001	2X	2	N	Y	I	N	Y
SVP001	51	2	N	Y	I	N	Y
FST001	58	2	N	Y	I	N	Y
TD0001	6X	2	N	Y	I	N	Y
EXOT01	7X	2	N	Y	I	N	Y

Table 24: Possible types for adding a controller with ADD-IO-UNIT (Part 1 of 2)

TYPE	DEV-TYP	CU	TA	SU	CP	DR	SV
EXOT02	7X	2	N	Y	S	N	Y
EXOT03	7X	2	Y	Y	S	N	Y
DISK01	8X , AX	2	N	Y	I	N	Y
DISK02	8X , AX	2	N	Y	S	N	Y
DISK03	8X , AX	2	N	N	S	N	Y
DISK04	8X , AX	2	N	N	S	Y	Y
DISK05	8X , AX	2	N	Y	S	Y	Y
TAPE01	BX , CX , EX	1	Y	Y	I	N	Y
TAPE02	BX , CX , EX	2	Y	Y	I	N	Y
TAPE03	BX , CX , EX	1	N	Y	I	N	Y
TAPE04	BX , CX , EX	2	N	Y	S	N	Y

Table 24: Possible types for adding a controller with ADD-IO-UNIT (Part 2 of 2)

The columns have the following meanings:

Column	Meaning
TYPE	Controller type
DEV-TYP	Device type code of the devices that can be connected to the controller type. The first character is the FAMILY code. If the second character is X, all devices of this device family can be connected.
CU	The control unit type indicates whether the controller can control more than one input/output simultaneously: 1 Type 1 controller controls only one input/output at any one time. 2 Type 2 controller controls several inputs/outputs simultaneously.
TA	Timeout Active specifies whether signal sequences between the channel and the controller are to be subject to timeout monitoring. Y The timeout is enabled. The maximum permissible delay time is 8 seconds. After this time has been exceeded the channel terminates the input/output concerned and generates an interrupt with "Interface Control Check". N The timeout is not enabled. No time monitoring by the channel takes place.

Table 25: Meaning of type when adding a controller with ADD-IO-UNIT (Part 1 of 2)

Column	Meaning
SU	<p>Suppress Data Feature specifies whether the channel can influence the data transfer from the peripherals to the controller in accordance with the load.</p> <p>Y Data transfer can be influenced depending on the load.</p> <p>N Data transfer cannot be influenced depending on the load.</p>
CP	<p>The Channel Protocol Type specifies which input/output interface protocol is used:</p> <p>I With the Interlock protocol, data requests have to be acknowledged.</p> <p>S The Streaming protocol operates with data rates defined by the controller.</p>
DR	<p>Indicates whether the controller supports the Dynamic Reconnection feature (dynamic channel reconnection).</p> <p>Y Dynamic channel reconnection is supported, i.e. the channel can be changes while input/output is underway.</p> <p>N Dynamic channel reconnection is not supported.</p>
SV	<p>Status Verification facility</p> <p>Y available</p> <p>N not available</p>

Table 25: Meaning of type when adding a controller with ADD-IO-UNIT (Part 2 of 2)

For disk and tape controllers on the Fibre Channel no distinction need be made between types. A disk controller can be generated with DISK01, a tape controller with TAPE02.

Example

The logical volumes with the LUNs 256 through 505 are configured on 2 ports of a disk controller with the port WWPNs 50060484360A4E8E and 50060484360A4E9A and connected to the two type FC channels with the channel numbers 38 and 3C (via an FC switch). The two controller ports are to be defined as a controller with the name 8800, the logical volumes as devices of the type D3435 with the names 8800 through 88F9. The logical volumes with LUN 256 is to be operated as a PAV volume with 8800 as the base device and 88FF as the alias device.

Procedure

1. Start configuration update:

```
/START-CONFIGURATION-UPDATE
```

2. Add channels:

```
/ADD-IO-UNIT UNIT=*CHANNEL(TYPE=*IBF(MODE=*FCP),CHANNEL-PATH-ID=38)
/ADD-IO-UNIT UNIT=*CHANNEL(TYPE=*IBF(MODE=*FCP),CHANNEL-PATH-ID=3C)
```

3. Add controller:

```
/ADD-IO-UNIT UNIT=*CONTR(NAME=8800,TYPE=DISK01,LOGICAL-ADDR=1,
                           PATH=( *CHANNEL(38,00,,50060484360A4E8E),
                                   *CHANNEL(3C,00,,50060484360A4E9A)),
                           DEV-RANGE=*DEV(00,256))
```

4. Add devices:

```
/ADD-IO-UNIT UNIT=*DEV(NAME=*RANGE(8800,250),TYPE=A5,
                        DEV-ADDR=00,PATH=*CONTR(8800)),
STATE=*DETACHED
```

5. Add PAV alias device:

```
/ADD-IO-UNIT UNIT=*DEV(NAME=88FF,TYPE=A5,DEV-ADDR=00,
                        PATH=*CONTR(8800),PAV-ALIAS-DEV=*YES(DEV-ADDR=FF)),
STATE=*DETACHED
```

6. Terminate configuration update:

```
/STOP-CONFIGURATION-UPDATE IORSF-UPDATE*YES(LEVEL=n)
```

ADD-ISAM-POOL-LINK

Assign pool link name to ISAM pool

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING

Function

The ADD-ISAM-POOL-LINK command is used to assign a pool link name to an ISAM pool for a user job and to enter this name in the task-local pool table. The job must already be linked to the specified ISAM pool (see the CREATE-ISAM-POOL command). With this pool link name, a connection can then be established via the TFT to a file, using the ADD-FILE-LINK command (with the operands LINK-NAME and POOL-LINK). The link name must be unique within a task. The user can have entries of the task-local pool table displayed by using the SHOW-ISAM-POOL-LINK command. Pool link names can be deleted from the pool table with the REMOVE-ISAM-POOL-LINK command.

For a full description of the ISAM pool see the “Introductory Guide to DMS” [13] or the “DMS Macros” manual [12].

Format

ADD-ISAM-POOL-LINK

```
LINK-NAME = <name 1..8>
,POOL-NAME = <name 1..8>(…)
  <name 1..8>(…)
    |
    | CAT-ID = *DEFAULT-PUBSET / <cat-id 1..4>
    |
    | ,SCOPE = *TASK / *HOST-SYSTEM / *USER-ID / *USER-GROUP
```

Operands

LINK-NAME = <name 1..8>

Link name by which the ISAM pool can be addressed (see the POOL-NAME operand). The link name must be unique within a task.

POOL-NAME = <name 1..8>(…)

Name of the ISAM pool to which the pool link name is to be assigned (see the LINK-NAME operand). The ISAM pool is uniquely identified by the specified name, the catalog ID (see the CAT-ID operand) and the scope (see the SCOPE operand). The job must already be linked to the ISAM pool (see the CREATE-ISAM-POOL or SHOW-ISAM-POOL-ATTRIBUTES command).

CAT-ID = *DEFAULT-PUBSET / <alphanum-name 1..4>

Catalog ID of the pubset to which the ISAM pool is assigned.

The specified catalog ID must match the specification in the corresponding CREATE-ISAM-POOL command.

CAT-ID = *DEFAULT-PUBSET

The ISAM pool is assigned to the catalog that was set with the ISPLDEFDC system parameter (ISAM-POOL-DEFAULT-CATID):

X'00': default catalog ID from the user entry (see the SHOW-USER-ATTRIBUTES command, output field *DEFAULT-PUBSET*)

X'01': catalog ID of the home pubset

CAT-ID = <alphanum-name 1..4>

Catalog ID of the pubset to which the ISAM pool is assigned.

SCOPE = *TASK / *HOST-SYSTEM / *USER-ID / *USER-GROUP

Specifies the scope of the ISAM pool.

The defined scope must match the specification in the corresponding CREATE-ISAM-POOL command.

SCOPE = *TASK

The pool link name is assigned to the task-local ISAM pool, i.e. the ISAM pool can only be used by its own task.

SCOPE = *HOST-SYSTEM

The pool link name is assigned to the cross-task ISAM pool (POOL-NAME operand), i.e. the ISAM pool can be used by all tasks.

SCOPE = *USER-ID / *USER-GROUP

These scopes are only supported for reasons of compatibility (see the CREATE-ISAM-POOL command).

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without error
	32	DMS0A17	Internal system error
	64	DMS0A0E	Syntax error in ISAM pool command
	64	DMS0A11	Specified catalog ID does not exist
	64	DMS0A13	Specified pool name is syntactically invalid
	64	DMS0A16	Pool link name already allocated
	64	DMS0A19	Specified ISAM pool does not exist
	64	DMS0A20	Parameters not permitted for remote BS2000 version
	64	DMS0A22	User group not present
	130	DMS0A12	Specified catalog ID not available

Examples

Use of an ISAM pool to process files

```

/show-isam-pool-attr pool=*all _____ (1)
%
% CATID    POOLNAME  SCOPE          WROUT    SIZE  EXTENTS  RESIDENT
%=====
% 10SU     SDFPOOLN  TASK          NO       128  --/--    NO
% 10SN     POOLAB01  HOST          YES      96   --/--    NO
% 10SN     POOLAB01  TASK          NO       96   --/--    NO
%
/add-isam-pool-link link=pool1,pool-name=poolab01(scope=*host) _____ (2)
/show-isam-pool-link pool-name=*all _____ (3)
%
% LINKNAME          CATID    POOLNAME      SCOPE
%=====
% SDFPOOLN          10SU     SDFPOOLN      TASK
% POOL1             10SN     POOLAB01      HOST
%
/show-file-attr abk.v100.isam,inf=(org=*yes) _____ (4)
%0000000126 :10SN:$USER1.ABK.V100.ISAM
%----- ORGANIZATION -----
% FILE-STRUC = ISAM          BUF-LEN = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE) = READ-WRITE    IO(PERF) = STD           DISK-WRITE = IMMEDIATE
% REC-FORM = (V,N)          REC-SIZE = 0
% KEY-LEN = 30              KEY-POS = 8
% AVAIL = *STD
% WORK-FILE = *NO          F-PREFORM = *K           SO-MIGR = *ALLOWED
%:10SN: PUBLIC: 1 FILE RES= 126 FRE= 5 REL= 3 PAGES
/add-file-link link=input1,file-name=abk.v100.isam,acc-method=*isam,
isam-attr=(key-pos=*by-cat,key-len=*by-cat,pool-link=pool1),
rec-form=*by-cat,buf-len=*by-cat,block-contr=*by-cat _____ (5)
/show-file-link inf=(file-contr=*yes) _____ (6)
%-- LINK-NAME ----- FILE-NAME -----
% INPUT1 :10SN:$USERXY01.ABK.V100.ISAM
%----- FILE-CONTROL-BLOCK - GENERAL ATTRIBUTES -----
% ACC-METH = ISAM          OPEN-MODE = *BY-PROG      REC-FORM = *BY-CAT
% REC-SIZE = *BY-PROG      BUF-LEN = *BY-CAT        BLK-CONTR = *BY-CAT
% F-CL-MSG = STD          CLOSE-MODE = *BY-PROG
%----- FILE-CONTROL-BLOCK - DISK FILE ATTRIBUTES -----
% SHARED-UPD = *BY-PROG    WR-CHECK = *BY-PROG      IO(PERF) = *BY-PROG
% IO(USAGE) = *BY-PROG     LOCK-ENV = *BY-PROG

```

```

% ----- FILE-CONTROL-BLOCK - TAPE FILE ATTRIBUTES -----
% LABEL      = *BY-PROG   (DIN-R-NUM = *BY-PROG, TAPE-MARK = *BY-PROG)
% CODE       = *BY-PROG   EBCDIC-TR  = *BY-PROG, F-SEQ    = *BY-PROG
% CP-AT-BLIM = *BY-PROG   CP-AT-FE0V = *BY-PROG  BLOCK-LIM = *BY-PROG
% REST-USAGE = *BY-PROG   BLOCK-OFF  = *BY-PROG  TAPE-WRITE = *BY-PROG
% STREAM     = *BY-PROG
% ----- FILE-CONTROL-BLOCK - ISAM FILE ATTRIBUTES -----
% KEY-POS    = *BY-CAT   KEY-LEN   = *BY-CAT   POOL-LINK = POOL1
% LOGIC-FLAG = *BY-PROG  VAL-FLAG  = *BY-PROG  PROPA-VAL = *BY-PROG
% DUP-KEY    = *BY-PROG  PAD-FACT  = *BY-PROG  READ-I-ADV = *BY-PROG
% WR-IMMED   = *BY-PROG  POOL-SIZE  = *BY-PROG

/start-exe from=*lib-elem(lib=ass.plamlib,elem=newwort3) _____ (7)
% BLS0517 MODULE 'NEUWORT3' LOADED
*** PRGLAUF OK! ***

/del-isam-pool pool=poolab01(scope=*task) _____ (8)

/del-isam-pool pool=poolab01(scope=*host) _____ (9)
% DMS0A1A POOL LINKS TO SPECIFIED POOL STILL EXIST. COMMAND NOT PROCESSED

/show-isam-pool-attr pool=*all _____ (10)
%
% CATID      POOLNAME  SCOPE          WROUT    SIZE  EXTENTS  RESIDENT
%=====
% 10SU      SDFPOOLN  TASK           NO       128  --/--    NO
% 10SN      POOLAB01  HOST           YES      96   2K/--    NO
%

/rem-isam-pool-link link=pool1 _____ (11)

/show-isam-pool-link pool-link=pool1 _____ (12)
% DMS0A60 SPECIFIED ISAM-POOL-LINK-NAME DOES NOT EXIST. COMMAND REJECTED

/del-isam-pool pool=poolab01(scope=*host) _____ (13)

/show-isam-pool-attr pool=*all _____ (14)
%
% CATID      POOLNAME  SCOPE          WROUT    SIZE  EXTENTS  RESIDENT
%=====
% 10SU      SDFPOOLN  TASK           NO       128  --/--    NO
%

```

- (1) Returns information on all ISAM pools connected to the task. There is one ISAM pool named *SDFPOOLN*, and one host-specific and one task-local ISAM pool with the name *POOLAB01*. It is evident that no file has been processed via these ISAM pools as yet, since the output column *EXTENTS* does not contain any formatting information.
- (2) Creates an entry in the task-local pool table. The pool link name *POOL1* is assigned to the host-specific ISAM pool *POOLAB01*.
- (3) Returns information on all pool link names.
- (4) Catalog information on the organization of the NK-ISAM file *ABK.V100.ISAM*.
- (5) Creates a TFT entry with the link name *INPUT1* for the file *ABK.V100.ISAM*. At the same time it also links the file with the pool link name *POOL1* (in the *POOL-LINK* operand).

- (6) Output of the TFT entry for the file *ABK.V100.ISAM* along with general information on file processing (FILE-CONTROL-BLOCK=*YES).
- (7) The program *NEWWORD3* is started from the library *ASS.PLAMLIB*. This program processes individual records of an ISAM file via an ISAM pool. The name of the file to be processed is obtained from the TFT entry with the link name *INPUT1*. After running successfully, the program terminates and writes the message **** PRGRUN OK! **** to SYSOUT.
- (8) The task-local ISAM pool *POOLAB01* is deleted.
- (9) The host-specific ISAM pool *POOLAB01* is to be deleted.. The DELETE-ISAM-POOL command is rejected, since there is still an entry in the task-local pool table for this pool (see step 10).
- (10) Returns information on all ISAM pools connected to the task. It is now evident that at least one NK2-ISAM file (here *ABK.V100.ISAM*) has been processed via the host-specific ISAM pool *POOLAB01*, since the ISAM pool is now formatted with 2K (see the *EXTENTS* column and step 1).
- (11) The entry with the pool link name *POOL1* is deleted.
- (12) There is now no longer an entry with the pool link name *POOL1*.
- (13) A new attempt is made to delete the host-specific ISAM pool *POOLAB01*.
- (14) The host-specific ISAM pool *POOLAB01* could be deleted. The task is now only connected to the task-local ISAM pool *SDFPOOLN*.

ADD-MASTER-CATALOG-ENTRY

Generate entry in MRSCAT of home pubset

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS
Routing code:	\$

Function

The command can be used to create MRSCAT entries both for single-feature pubsets (SF pubsets) and for system-managed pubsets (SM pubsets). The newly entered catalog is assigned the status “inaccessible”. An entry is not created if one already exists for the specified catalog ID. The catalog IDs must be unique throughout the entire network, i.e. the disks must be initialized accordingly using the VOLIN utility routine.

Unlike the IMPORT-PUBSET and EXPORT-PUBSET commands, this command does not affect the accessibility of the catalogs. With single-feature pubsets, default values are assigned for all cache attributes; with system-managed pubsets they are assigned for the cache attributes FORCE-IMPORT and SIZE-TOLERANCE only. If attributes need to be modified later, the MODIFY-PUBSET-CACHE-ATTRIBUTES command must be used (see the “Introduction to System Administration” [14]; for administration of SM pubsets see also “System-Managed Storage” [45]).

This command cannot be used to create a volume set entry. Volume set entries are generated automatically by the system when required.

Format

(Part 1 of 2)

ADD-MASTER-CATALOG-ENTRY

ENTRY-NAME = <cat-id 1..4>

,**PUBSET-TYPE** = ***SINGLE-FEATURE** (...) / ***SYSTEM-MANAGED**(...)

***SINGLE-FEATURE**(...)

START-SPEEDCAT = ***AUTOMATIC** / ***NO** / ***SPEEDCAT-TASK** / ***OWN-TASK**

PHYSICAL-ALLOCATION = ***ADMINISTRATOR-ONLY** / ***USER-ALLOWED**

```

,ALLOCATION = *STD / *PARAMETERS(...)
  *PARAMETERS(...)
    SATURATION-LEVEL4 = *STD / <integer 66..2147483647 2Kbyte>
    ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215 2Kbyte>
    ,SECONDARY-ALLOCATION = *STD / <integer 1..32767 2Kbyte>
    ,MAXIMAL-ALLOCATION = *STD / <integer 1..32767 2Kbyte>
  *SYSTEM-MANAGED(...)
    CONTROL-VOLUME-SET = *NONE / <cat-id 1..4>
  ,PARTNER-NAME = *OWN / <alphanum-name 1..8>
  ,ACCESS-FAILURE = *HOLD-JOBS / *CANCEL-JOBS
  ,RESIDENT-BUFFERS = *SYSTEM-STD / *NO / *YES
  ,NUMBER-OF-BUFFERS = *SYSTEM-STD / <integer 1..255>
  ,BATCH-WAIT-TIME = 28800 / <integer 0..2147483647 seconds>
  ,DIALOG-WAIT-TIME = 30 / <integer 0..2147483647 seconds>
  ,SHARED-PUBSET = *NO / *YES
  ,ACCESS-CONTROLLED = *NO / *YES(...)
  *YES(...)
    USER-IDENTIFICATION = *TSOS / <alphanum-name 1..8>
  ,EAM = *STD / *PARAMETERS(...)
  *PARAMETERS(...)
    MAXIMAL-SIZE = *STD / <integer 12..193536 2Kbyte>
    ,MINIMAL-SIZE = *STD / <integer 12..193536 2Kbyte>
    ,SECONDARY-ALLOCATION = *STD / <integer 1..193536 2Kbyte>
    ,VIRTUAL-MEMORY = *STD / <integer 0..8192 2Kbyte>
  ,REMOTE-IMPORT = *BY-CONNECTION / *BY-COMMAND-ONLY
  ,XCS-CONFIGURATION = *NO / *YES
  ,PUBRES-UNIT = *STD / <alphanum-name 2..2> / <x-text 4..4>

```

Operands**ENTRY-NAME = <cat-id 1..4>**

Catalog ID of the pubset for which a new entry is to be generated in the MRSCAT.

PUBSET-TYPE = *SINGLE-FEATURE(...) / *SYSTEM-MANAGED(...)

Indicates the type of pubset for which a new entry is to be generated in the MRSCAT.

PUBSET-TYPE = *SINGLE-FEATURE(...)

The pubset for which a new entry is to be generated in the MRSCAT is an SF pubset.

START-SPEEDCAT = *AUTOMATIC / *NO / *SPEEDCAT-TASK / *OWN-TASK

Governs whether or not SCA is to be available for this pubset.

START-SPEEDCAT = *AUTOMATIC

If the SPEEDCAT function is available, SCA is started automatically when the pubset is imported under a local task, the SPEEDCAT task.

START-SPEEDCAT = *NO

SCA is not to be available.

START-SPEEDCAT = *SPEEDCAT-TASK

SCA is to be available, and should run under a separate task, the SPEEDCAT task.

START-SPEEDCAT = *OWN-TASK

SCA should be available and run under the user's task.

PHYSICAL-ALLOCATION = *ADMINISTRATOR-ONLY / *USER-ALLOWED

Specifies whether or not users may directly allocate individual volumes of this pubset.

PHYSICAL-ALLOCATION = *ADMINISTRATOR-ONLY

Users may not directly allocate the volumes of this pubset. This right is reserved for the privileged caller under TSOS.

PHYSICAL-ALLOCATION = *USER-ALLOWED

On this pubset, nonprivileged users will also be permitted to allocate individual volumes.

ALLOCATION = *STD / *PARAMETERS(...)

Specifies whether or not particular allocation parameters should be set for data processing on the pubset.

ALLOCATION = *STD

The setting *STD, i.e. the value of the system parameter concerned (see the description of the *STD operand values for the various allocation parameters below), should be specified for all allocation parameters.

ALLOCATION = *PARAMETERS(...)

Specifies allocation parameters for the pubset, with the values which follow.

SATURATION-LEVEL4 = *STD / <integer 66..2147483647 2Kbyte>

The value specified in the system parameter L4SPDEF is to apply.

SATURATION-LEVEL4 = *STD

The standard value established at system generation (class 2 system parameter L4SPDEF) is to apply.

SATURATION-LEVEL4 = <integer 66..2147483647 2Kbyte>

Value of memory saturation level 4 which is to apply to this pubset.

PRIMARY-ALLOCATION = *STD / <integer 1..16777215 2Kbyte>

The allocation parameter for the primary allocation when a storage space request is made (file processing) is to be set.

PRIMARY-ALLOCATION = *STD

The value specified in the system parameter DMPRALL is to apply.

PRIMARY-ALLOCATION = <integer 1..16777215 2Kbyte>

The value to be applied for this pubset, for primary storage space allocations during file processing. This value will apply as the default value for any user who does not specify an initial allocation in a storage space request.

SECONDARY-ALLOCATION = *STD / <integer 1..32767 2Kbyte>

The allocation parameter for secondary allocations when a storage space request is made (file processing) is to be set.

SECONDARY-ALLOCATION = *STD

The value specified in the system parameter DMSCALL is to apply.

SECONDARY-ALLOCATION = <integer 1..32767 2Kbyte>

The value to be applied for this pubset, for secondary storage space allocations during file processing. This value will apply as the default value for any user who does not specify an initial allocation in a storage space request.

MAXIMAL-ALLOCATION = *STD / <integer 1..32767 2Kbyte>

The allocation parameter for doubling the secondary allocation when a storage space request is made (file processing) is to be set.

MAXIMAL-ALLOCATION = *STD

The value specified in the system parameter DMMAXSC is to apply.

MAXIMAL-ALLOCATION = <integer 1..32767 2Kbyte>

The value assigned to this pubset for doubling of the secondary storage space allocation during file processing. This value will be the maximum value for any user, for whom the space remains inadequate even after a secondary allocation has been made. The value for the secondary allocation will be repeatedly doubled until the specified maximum value is reached.

PUBSET-TYPE = *SYSTEM-MANAGED(...)

The pubset for which a new entry is to be generated in the MRSCAT is an SM pubset.

CONTROL-VOLUME-SET = *NONE / <cat-id 1..4>

Governs whether a control volume set is used.

CONTROL-VOLUME-SET = *NONE

A control volume set is not to be used.

CONTROL-VOLUME-SET = <cat-id 1..4>

Specifies the catalog ID of the pubset that is to be used as the control volume set.

PARTNER-NAME = *OWN / <alphanum-name 1..8>

BCAM name of the partner processor for remote file access.

PARTNER-NAME = *OWN

No BCAM name is specified.

PARTNER-NAME = <alphanum-name 1..8>

BCAM name of the processor for remote file access (RFA). A processor can only be specified if no MRS connection exists but RFA is nevertheless required.

If an MSCF connection exists and REMOTE-IMPORT=*BY-CONNECTION is specified, the system automatically enters the name of the master computer.

ACCESS-FAILURE = *HOLD-JOBS / *CANCEL-JOBS

The specification is still accepted for reasons of compatibility and also stored in the MRSCAT. However, the system no longer evaluates this setting.

The system behavior when a connection to a pubset is lost is selected using the BATCH-WAIT-TIME or DIALOG-WAIT-TIME operand.

RESIDENT-BUFFERS = *SYSTEM-STD / *NO / *YES

Specifies whether resident or nonresident buffers are to be created.

RESIDENT-BUFFERS = *SYSTEM-STD

The system automatically decides which memory class to use.

RESIDENT-BUFFERS = *NO

Non-resident buffers are used.

RESIDENT-BUFFERS = *YES

Resident buffers are used.

NUMBER-OF-BUFFERS = *SYSTEM-STD / <integer 1..255>

Defines the number of buffers.

The buffer specification is evaluated - according to the following hierarchy - only when the pubset is imported:

1. Explicit parameter entry in the IMPORT-PUBSET command.
2. Information on the ADD- or MODIFY-MASTER-CATALOG-ENTRY command.

If only one of the operands RESIDENT-BUFFERS or NUMBER-OF-BUFFERS is specified, the default value is assumed for the other (RESIDENT-BUFFERS=*NO, NUMBER-OF-BUFFERS=32).

3. Specifications according to the system parameters CATBUFR and BMTNUM.

BATCH-WAIT-TIME = 28800 / <integer 0..2147483647 seconds>

Time in seconds that batch jobs are to wait for the availability of pubsets that cannot be accessed due to a connection failure. If the time specified is exceeded, command processing is aborted with an error code and the spin-off mechanism is triggered. The default value for this wait time is 28800 seconds (8 hours).

DIALOG-WAIT-TIME = 30 / <integer 0..2147483647 seconds>

As with batch jobs, defines wait time for interactive jobs. If the time specified is exceeded, the interactive job receives a message and is resumed. The default value here is 30 seconds.

SHARED-PUBSET = *NO / *YES

Determines the shareability of the pubset for local processors.

ACCESS-CONTROLLED = *NO / *YES

Specifies whether access to the pubset is to be restricted to one user ID.

ACCESS-CONTROLLED = *NO

Access to the pubset is not subject to any restrictions.

ACCESS-CONTROLLED = *YES(...)

Access to the pubset is to be restricted to one user ID.

USER-IDENTIFICATION = *TSOS / <alphanum-name 1..8>

Specifies the user ID which is to have access to the pubset.

USER-IDENTIFICATION = *TSOS

Only the TSOS user ID has access to the pubset.

USER-IDENTIFICATION = <alphanum-name 1..8>

Specifies a user ID, in addition to the TSOS user ID, which is to be permitted access to the pubset.

EAM = *STD / *PARAMETERS(...)

Specifies the SYSEAM file parameters specific to this pubset.

EAM = *STD

The setting *STD, i.e. the value of the system parameter concerned (see the description of the *STD operand values for the various SYSEAM parameters below), should be specified for all SYSEAM parameters.

EAM = *PARAMETERS(...)

Specifies the SYSEAM parameters which follow for the pubset.

MAXIMAL-SIZE = *STD / <integer 12..193536 2Kbyte>

This operand is only supported for reasons of compatibility and is no longer evaluated.

MINIMAL-SIZE = *STD / <integer 12..193536 2Kbyte>

Specifies the minimum size (number of half pages) that the SYSEAM file may occupy in the specified pubset.

MINIMAL-SIZE = *STD

The value specified in the system parameter EAMMIN is to apply.

MINIMAL-SIZE = <integer 12..193536 2Kbyte>

Minimum size (number of half pages) that the SYSEAM file may occupy in the pubset.

SECONDARY-ALLOCATION = *STD / <integer 1..193536 2Kbyte>

Specifies the secondary allocation for the SYSEAM file, for this pubset.

SECONDARY-ALLOCATION = *STD

The value specified in the system parameter EAMSEC is to apply.

SECONDARY-ALLOCATION = <integer 1..193536 2Kbyte>

Value of the secondary allocation for the SYSEAM file (number of half pages). This value should be a multiple of 24.

VIRTUAL-MEMORY = *STD / <integer 0..8192 2Kbyte>

Specifies the number of half pages to be used in determining the size of the EAM cache area in class 4 memory. A cache area can only be created for the SYSEAM file of the home pubset. The parameter has no effect for any other pubsets.

VIRTUAL-MEMORY = *STD

The size of the EAM cache area will be determined from the value of the class 2 system parameter EAMMEM.

VIRTUAL-MEMORY = <integer 0..8192 2Kbyte>

The number of half pages specified here determines the size of the EAM cache area.

REMOTE-IMPORT = *BY-CONNECTION / *BY-COMMAND-ONLY

Defines how the remote import status can be changed.

REMOTE-IMPORT = *BY-CONNECTION

The change is made on connection setup.

REMOTE-IMPORT = *BY-COMMAND-ONLY

The remote import status can only be changed by a command.

XCS-CONFIGURATION = *NO / *YES

For shared pubset only: Defines whether the pubset may be automatically imported as an XCS pubset when a new XCS network is established.

XCS-CONFIGURATION = *NO

The pubset must not be imported automatically as an XCS pubset. (The pubset may nonetheless be used as an XCS pubset using the SET-XCS-PUBSET command.)

XCS-CONFIGURATION = *YES

The pubset is to be imported automatically as an XCS pubset by the MSCF subsystem when a new XCS network is established.

PUBRES-UNIT = *STD / <alphanum-name 2..2> / <x-text 4..4>

Mnemonic device name (MN) of the pubres of the SF pubset or of the volres of the SM pubset. VM2000 requires this entry when the pubset is specified by means of its catalog ID.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without error
2	0	CMS0002	Disk error
	1	CMS0011	Syntax error
	1	CMS0314	Syntax error in entry name or error in wildcard specification
	32	CMD0221	Internal system error
	32	CMS0001	SLOT manager error
	32	CMS031F	MRSCAT parameter error
	32	CMS0310	Error during privilege checking
	32	CMS0317	MRSCAT is locked
	32	CMS0318	Synchronization error due to task lock manager problem
	64	CMS0004	MRSCAT entry already exists
	64	CMS0005	Too many entries in the MRSCAT
	64	CMS0010	Command reserved for systems support

Examples

Example of creating entries in the local catalog and import pubsets (MPVS)

The pubset MAX is also to be imported locally on the system with the HOME pubset FOR, and a paging file already contained on the pubset PAG is to be used. Furthermore, the pubset LUX, which is imported on the remote system XY, should be accessible via RFA.

Assuming that the disks belonging to pubsets MAX and PAG are already available, the following commands must be entered:

```
/ADD-MASTER-CATALOG-ENTRY MAX
/ADD-MASTER-CATALOG-ENTRY PAG
/ADD-MASTER-CATALOG-ENTRY LUX, PARTNER-NAME=XY
/IMPORT-PUBSET MAX
/IMPORT-PUBSET PAG
/EXTEND-PAGING-AREA PAG.00
/EXPORT-PUBSET PAG
```

The catalog directory MRSCAT in the home pubset has the following contents:

```
PUBSET FOR: LOCAL-HOME, PAGING
PUBSET LUX: INACC, HOST=XY
PUBSET MAX: LOCAL-IMPORTED
PUBSET PAG: INACC, PAGING
```

Example of creating catalog entries in a computer network (HIPLEX MSCF)

A computer network is to be set up for single-computer systems HOST1 and HOST2 with catalog IDs A and B. Each system has a MSCF configuration file containing an entry for the other system. The catalog directories contain the following information:

HOST1
A,HOST1,LOCAL

HOST1

HOST2
A,INACC B,HOST2,LOCAL

HOST2

- 1) /START-SUBSYSTEM SUBSYS = MSCF
- 2) /ADD-MAS-CAT-ENTRY ENTRY=B

- 3) /STA-SUBSYSTEM SUBSYS = MSCF

Thus when the command START-SUBSYSTEM SUBSYS=MSCF is issued, a MSCF connection is set up between the two systems and the MRSCAT is updated.

If the commands are given in the order

1) - 2) - 3)

then pubset B is accessible to HOST1 since the ADD-MASTER-CATALOG-ENTRY command was issued before HOST1 and HOST2 were connected.

If the commands are given in the order

1) - 3) - 2)

the connection between HOST1 and HOST2 is established before the ADD-MASTER-CATALOG-ENTRY command is issued. In this case pubset B is not accessible to HOST1. Updating of this MRSCAT entry is not performed until the next time a connection is set up (recovery, STOP-/START-MSCF-CONNECTION, STOP-/START-SUBSYSTEM MSCF). If, however, this catalog is to be accessed immediately, the following command must be issued.

```
/IMPORT-PUBSET PUBSET=B, USE=*FROM-REMOTE(HOST-NAME=HOST2)
```

Note

If the two systems had no configuration files, the command /START-SUBSYSTEM SUBSYS=MSCF would not cause an MSCF connection to be set up and would not update the MRSCAT. MSCF would be loaded, but the systems would remain isolated.

ADD-NET-STORAGE-VOLUME

Create Net-Storage volume and assign it to a local pubset

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Net-Storage administration
Domain:	STORAGE-MANAGEMENT
Privileges:	TSOS

Function

The ADD-NET-STORAGE-VOLUME command creates a directory with management files (catalog and FSL files) in the released file system of the net server and establishes the unambiguous assignment of the directory to precisely one local SF or SM pubset in the form of a Net-Storage volume. The pubset must already have been imported. No Net-Storage volume can be assigned to the home pubset.

From the viewpoint of BS2000 a Net-Storage volume is a new disk which is assigned to the pubset but which is not (as in the case of a pubset expansion) added to the pubset. The VSN of the Net-Storage volume is the same as the directory name in the Net-Storage and by default is derived from the name of the pubset, see the "Introduction to System Administration" [14]. Only if the pubset has already been assigned a Net-Storage volume with a default name must a VSN which complies with the conventions for private volumes be specified.

The IMPORT operand enables a directory which already exists on the net server to be assigned to the specified pubset as a Net-Storage volume. Renaming (VSN or directory name) is also possible.



After the Net-Storage volume has been created, any files it may contain must be imported using the IMPORT-FILE command before they can be accessed.

Information on the Net-Storage available in BS2000 can be requested using the SHOW-NET-STORAGE command. Information on the Net-Storage available in the pubset can be requested using the SHOW-PUBSET-NET-STORAGE command.

Fundamental information on the use of Net-Storage in BS2000 is provided in the "Introduction to System Administration" [14]. How to work with files on Net-Storage is described in the "Introductory Guide to DMS" [13].

Format

ADD-NET-STORAGE-VOLUME
<pre>VOLUME = *STD / <vsn 6..6> , PUBSET = <cat-id 1..4> , DIRECTORY = <composed-name 1..64 with-under> / <c-string 1..64 with-low> , SERVER = <composed-name 1..256 with-under> / <c-string 1..256 with-low> / *IP-ADDRESS(...) *IP-ADDRESS(...) IP-ADDRESS = <composed-name 7..15> / <c-string 2..39> , IMPORT = *NO / *YES(...) *YES(...) OLD-VOLUME-NAME = *SAME / <vsn 6..6> FORCE = *NO / *YES</pre>

Operands

VOLUME =

Specifies the VSN of the Net-Storage volume.

On the Net-Storage the Net-Storage volume is implemented by a directory below the released directory with a name which corresponds to the VSN.

VOLUME = *STD

The VSN of the new Net-Storage volume is derived from the name of the pubset, see the “Introduction to System Administration” [14]. If the pubset has already been assigned a Net-Storage volume with a default name, this specification is not possible and the VSN must be specified explicitly.

VOLUME = <vsn 6..6>

Specifies the VSN of the Net-Storage volume explicitly. A default name cannot be specified. The VSN must consist of 6 characters and comply with the conventions for private volumes. Consequently it may not begin with the string PUB and may not contain a period, see the “Introduction to System Administration” 14].

PUBSET = <cat-id 1..4>

Specifies the name of the pubset to which the Net-Storage volume will be assigned.

DIRECTORY = <composed-name 1..64 with-underscore> / <c-string 1..64 with-low>

Specifies the directory name of the Net-Storage released on the net server.

On the Net-Storage the Net-Storage volume is implemented by a directory below the released directory with a name which corresponds to the VSN.

SERVER =

Specifies the net server which makes the Net-Storage available. Either the host name, the fully qualified domain name or the IP address of the net server is specified.

SERVER = <composed-name 1..256 with-underscore> / <c-string 1..256 with-low>

Host name or fully qualified domain name of the net server.

SERVER = *IP-ADDRESS(...)

IP address of the net server.

IP-ADDRESS = <composed-name 7..15> / <c-string 2..39>

Specifies the IP address in IPv4 or IPv6 format.

IMPORT =

Specifies whether a new directory is to be created for the specified volume or whether the directory already exists on the Net-Storage.

IMPORT = *NO

Creates a new directory on the Net-Storage for the specified volume and assigns it to the specified pubset as a Net-Storage volume.

IMPORT = *YES(...)

A directory already exists on the Net-Storage for the specified volume. This is assigned to the specified pubset as a Net-Storage volume. The catalog on the Net-Storage is **not** imported here. As a result the files on the Net-Storage can be accessed only after they have been imported with IMPORT-FILE.

OLD-VOLUME-NAME =

Specifies whether the VSN of the Net-Storage volumes is to be retained.

OLD-VOLUME-NAME = *SAME

The Net-Storage volume already has the VSN specified in the VOLUME operand. It need not be renamed.

OLD-VOLUME-NAME = <vsname 6..6>

The Net-Storage volume has so far had the VSN specified here and is to be renamed. In this case the directory which already exists on the Net-Storage and the Net-Storage volume are respectively assigned the name and the VSN which is specified in the VOLUME operand.

FORCE = *NO / *YES

Specifies whether the Net-Storage volume is to be assigned to the pubset even if inconsistencies still exist (e.g. not properly released).

The default *NO causes the command to be rejected in this case.

FORCE = *YES

In all cases the Net-Storage volume is assigned to the pubset. Any inconsistencies which still exist are ignored.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed without error
	32	CMD0221	system error
	32	DMS1CFF	System error (see SERSLOG entry)
	64	DMS1C04	Error when calling the ONETSTOR subsystem
	64	DMS1C05	Catalog ID does not exist
	64	DMS1C06	Pubset not imported
	64	DMS1C07	Error when calling the allocation function
	64	DMS1C08	Error when calling a CMS function for the TSOSCAT
	64	DMS1C09	Error when calling a CMS function on the Net-Storage volume
	64	DMS1C0A	Catalog entry of the Net-Storage volume already exists
	64	DMS1C0B	File system label already exists on the net server
	64	DMS1C0D	File system label on the net server does not match the pubset
	64	DMS1C0F	No new Net-Storage volume may be created on the slave system
	64	DMS1C11	The Net-Storage volume already exists
	64	DMS1C12	No Net-Storage volume can be created on the home pubset.
	64	DMS1C17	Net directory cannot be accessed
	64	DMS1C1A	Net-Storage volume already placed in service
	64	DMS1C20	Name of the Net-Storage volume is invalid
	64	DMS1C21	Incorrect syntax in the IP address
	64	DMS1C0F	Command not permitted on the slave

ADD-PASSWORD

Add password to password table for job

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE FILE-GENERATION-GROUP JOB-VARIABLES
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION
Routing code:	\$ (with NBCONOPI=N) or E (with NBCONOPI=Y)

Function

The ADD-PASSWORD command supplies a job with the passwords needed to access password-protected files and job variables. A password table is created for the job, and the specified passwords are entered into it. When a protected file or job variable is to be accessed, the system searches the password table for the required password. If the appropriate password is found, access is granted. In commands which access files or job variables, the required password can be specified directly with the command in some cases (e.g. when deleting files or job variables, starting up programs, etc). Entering the correct password, however, will only enable a single access during command execution. In programs the password can also be specified in the file control block (FCB). The password table is deleted when the job terminates. Individual entries or all entries in it can be optionally deleted from the table before the job ends by using the REMOVE-PASSWORD command.

Note

Only one entry is created in the password table of the job for each password. In other words, if an ADD-PASSWORD command is issued for a password that already exists in the password table, no new entry will be created in the password table. Existing entries can be deleted with the REMOVE-PASSWORD command. Additional REMOVE-PASSWORD commands for deleted passwords are rejected.

In outputs such as job logs, the passwords are not shown in plain text. Where a password has been specified, it is represented by the letter 'P'.

Protection by password can be extended by internal encoding of the passwords, carried out by the system. They then only appear in coded form in memory dumps, for example. Password encoding is set up by the system parameter ENCRYPT.

Password protection for files is dealt with in the “Introductory Guide to DMS” [13]; for details of password protection for job variables, see the “Job Variables” [20] manual.

REMOTE-FILE-ACCESS

The command ADD-PASSWORD is automatically passed on by the requesting job to all processes participating in RFA (see the “RFA” manual [31]).

Format

ADD-PASSWORD	Alias: ADPW
PASSWORD = *SECRET / list-poss(63): <x-string 1..8> / <c-string 1..4> / <integer -2147483648..2147483647>	

Operands

**PASSWORD = *SECRET / list-poss(63): <x-string 1..8> / <c-string 1..4> /
<integer -2147483648..2147483647>**

Passwords which are to be entered in the password table. A maximum of 63 passwords may be specified by one ADD-PASSWORD command.

The operand has the following special characteristics:

- The input field is automatically blanked out in the guided dialog.
- In unguided dialog and foreground procedures, the entry *SECRET or ^, SDF provides a blanked out input field for inputting the password .
- The password entered is not logged.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
	1	CMD0202	Syntactical or semantic error in command
	32	DMS0584	A state that does not allow the function to continue was reported during processing
	64	DMS0691	Password table at maximum size
	64	DMS0692	Maximum number of passwords per task reached
	64	DMS06FF	BCAM connection severed
	130	DMS0585	Error detected during catalog processing or multihost processing
	130	DMS0594	Not enough virtual memory available

The following applies to the specification of passwords:

A file can be protected against read access, write access or execute access by three different passwords. Changes to the catalog entry and deletion of the file are classified as write accesses. This also applies to job variables, but only for read and write access, i.e. only a read password and write password are possible.

The table below shows the possible combinations of password protection, and which passwords are required for any particular type of access:

Password protection	Password specified	Permitted access type
EXEC-PASSWORD	None specified	No access
	Execution password	Execute Read Write
READ-PASSWORD	None specified	Execute *)
	Read password	Execute Read Write
WRITE-PASSWORD	None specified	Execute Read
	Write password	Execute Read Write

Table 26: Password specifications for each type of access (Part 1 of 2)

Password protection	Password specified	Permitted access type
EXEC-PASSWORD READ-PASSWORD WRITE-PASSWORD	None specified	No access
	Execution password	Execute *)
	Read password	Execute Read
	Write password	Execute Read Write
EXEC-PASSWORD READ-PASSWORD	None specified	No access
	Execution password	Execute *)
	Read password	Execute Read Write
EXEC-PASSWORD WRITE-PASSWORD	None specified	No access
	Execution password	Execute Read
	Write password	Execute Read Write
READ-PASSWORD WRITE-PASSWORD	None specified	Execute *)
	Read password	Execute Read
	Write password	Execute Read Write

*) The program code is protected against dump access

Table 26: Password specifications for each type of access (Part 2 of 2)

Example*Accessing files with read, write, and execute passwords*

```

/mod-file-attr proc.mini.1,prot=(read-pass='john',
                               write-pass='paul',exec-pass='maxi') ----- (1)
/call-proc proc.mini.1,log=*yes ----- (2)
% SDP0094 CONTAINER NOT ACCESSIBLE
% SDP0093 ERROR DURING ACCESS OF FILE/LIBRARY ':20SG:$USERXY01.PROC.MINI.1',
  ERROR 'DMS0D91'. MORE INFORMATION: /HELP-MSG DMS0D91
% SDP0094 CONTAINER NOT ACCESSIBLE
/add-pass 'maxi' ----- (3)
/call-proc proc.mini.1,log=*yes ----- (4)
% SDP0224 LOGGING SUPPRESSED; CONTAINER ':20SG:$USERXY01.PROC.MINI.1' IS
  READ PROTECTED
** The time now is: 15:33:54 **
/print-doc proc.mini.1 ----- (5)
% SCP0860 FILE ':20SG:$USERXY01.PROC.MINI.1' PROTECTED BY A READ OR EXEC
  PASSWORD. COMMAND REJECTED
/rem-pass 'maxi' ----- (6)
/add-pass 'johnjohn' ----- (7)
/call-proc proc.mini.1,log=*yes ----- (8)
%      1 1 /WRITE-TEXT '** The time now is: 15:35:45 **'
** The time now is: 15:35:45 **
%      1 /EXIT-PROCEDURE ERROR=*NO
/mod-f-attr proc.mini.1,proc.mini.2 ----- (9)
% DMS0681 DMS ERROR '05CF' WHEN ACCESSING FILE ':20SG:$USERXY01.PROC.MINI.1'.
  FOR FURTHER INFORMATION: /HELP-MSG DMS05CF
/help-msg dms05cf ----- (10)
% DMS05CF FILE PROTECTED BY (CRYPTO)PASSWORD. ADD REQUIRED (CRYPTO)PASSWORD
  AND REENTER COMMAND
% ? The user tried to modify the catalog entry of a file that is
%   protected by a (crypto)password, but the required (crypto)password
%   has not yet been added to the (crypto)password table of the task.
% ! Add (crypto)password to the (crypto)password table
%   (command ADD-(CRYPTO-)PASSWORD) and reenter command.
/rem-pass 'john' ----- (11)
/add-pass 'paul' ----- (12)
/mod-f-attr proc.mini.1,proc.mini.2 ----- (13)
/call-proc proc.mini.2,log=*yes ----- (14)
%      1 1 /WRITE-TEXT '** The time now is: 15:37:42 **'
%      1 /EXIT-PROCEDURE ERROR=*NO

```

- (1) The file *PROC.MINI.1* is protected by the read password *JOHN*, the write password *PAUL*, and the execute password *MAXI*. The file contains an S procedure which outputs the current time to SYSOUT.
- (2) The procedure *PROC.MINI.1* is called. The CALL-PROCEDURE command is rejected, since the required execution password is not contained in the password table of the task.
- (3) Entry of the execution password *MAXI* into the password table.
- (4) The repeated call to the procedure succeeds, but the procedure is not logged on SYSOUT, since the password required for read access is not contained in the password table.
- (5) The PRINT-DOCUMENT command is also rejected due to the missing read password.
- (6) The execute password *MAXI* is deleted from the password table.
- (7) Entry of the read password *JOHN* into the password table.
- (8) The call to the procedure succeeds, and the processing sequence is logged on SYSOUT. The read password provides authorization for reading as well as execution.
- (9) The file *PROC.MINI.1* is to be renamed. The MODIFY-FILE-ATTRIBUTES command is rejected.
- (10) The HELP-MSG-INFORMATION command returns information on the cause of the error: the password required for write access (to change the catalog entry in this case) is not contained in the password table.
- (11) The read password *JOHN* is deleted from the password table.
- (12) Entry of the write password *PAUL* into the password table.
- (13) The file *PROC.MINI.1* can now be renamed to *PROC.MINI.2* (write access).
- (14) The call to the procedure under the new name *PROC.MINI.2* succeeds, and the processing sequence is logged on SYSOUT. The write password provides authorization for read, write, and execute access.

ADD-POSIX-USER

Define POSIX user attributes

Description status:	POSIX-BC V10.0A
Functional area:	User management POSIX administration and application
Domain:	SYSTEM-MANAGEMENT
Privileges:	TSOS USER-ADMINISTRATION

This command can only be used if the chargeable subsystem SDF-P is loaded.

Function

The ADD-POSIX-USER command defines all the POSIX attributes of the BS2000 user ID for a new POSIX user. The necessary preparations are also made in POSIX for this user to permit POSIX access (creating the login directory for this user).

Note

This command replaces the S procedure POSADDUS.

The POSIX user attributes of a BS2000 user ID can be modified using the MODIFY-POSIX-USER-ATTRIBUTES command. Information on the current POSIX default attributes can be obtained with the SHOW-POSIX-USER-DEFAULTS command.

Format

ADD-POSIX-USER

```

USER-NAME = <name 1..8>
, USER-NUMBER = *DEFAULT / <integer 0..60002>
, GROUP-NUMBER = *DEFAULT / <integer 0..60002>
, PROGRAM = *DEFAULT / <posix-pathname 1..1023 without-wild>
, HOME-DIRECTORY = *DEFAULT / <posix-pathname 1..1023 without-wild>
, RLOGIN-ACCOUNT = *NONE / <alphanum-name 1..8>

```

Operands

USER-NAME = <name 1..8>

BS2000 user ID whose POSIX user attributes are to be defined.

USER-NUMBER =

User number which is to be defined for this BS2000 user ID.

The USER-NUMBER attribute is relevant to security as the user number indicates the privilege and determines the owner of a file.

USER-NUMBER = *DEFAULT

The user number is given the currently set default value (see the SHOW-POSIX-USER-DEFAULTS command).

USER-NUMBER = <integer 0..60002>

The user number is given the specified value.

GROUP-NUMBER =

Group number which is to be defined for the BS2000 user ID.

The GROUP-NUMBER attribute is relevant to security as POSIX does not check the permissibility of the combination of BS2000 user ID and group against the POSIX group catalog when you log on.

GROUP-NUMBER = *DEFAULT

The group number is given the currently set default value (see the SHOW-POSIX-USER-DEFAULTS command).

GROUP-NUMBER = <integer 0..60002>

The group number is assigned the specified value.

PROGRAM =

Program which is started after the *rlogin* command or after the START-POSIX-SHELL command is called.

This attribute is not relevant to security as only programs which the user is allowed to execute are started.

PROGRAM = *DEFAULT

The program to be started is determined on the basis of the currently set default value (see the SHOW-POSIX-USER-DEFAULTS command).

PROGRAM = <posix-pathname 1..1023 without-wild>

The specified program is started.

HOME-DIRECTORY =

Determines the absolute path name of the directory which the user is automatically directed to (login directory) after connecting with POSIX.

This attribute is not relevant to security as it only determines the content of the shell variable HOME and the initial value of the working directory. The protection attributes of files and directories cannot be bypassed in this way.

If the directory does not yet exist it is created and the owner is set to the user number and the group number of the POSIX user ID.

If the directory already exists its attributes remain unchanged and a respective message is output.

HOME-DIRECTORY = *DEFAULT

The directory determined on the basis of the currently set default value (see the SHOW-POSIX-USER-DEFAULTS command).

HOME-DIRECTORY = <posix-pathname 1..1023 without-wild>

Specifies the directory.

RLOGIN-ACCOUNT =

Determines the account number for POSIX access via remote login or NFS.

RLOGIN-ACCOUNT = *NONE

No account number is specified. The account number defined in the home pubset user entry thus remains unchanged for POSIX access.

RLOGIN-ACCOUNT = <alphanum-name 1..8>

The specified account number is entered in the home pubset user entry as the new account number for POSIX access (see the POSIX-RLOGIN-DEFAULT operand in the ADD-USER and MODIFY-USER-ATTRIBUTES commands).

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
	64	SDP0018	An error has occurred in the S procedure called by the command. Information on the precise cause is provided by one of the messages POS2900 through POS2905.
	65	CMD2241	The chargeable subsystem SDF-P is not available.

ADD-SUBSYSTEM

Extend dynamic subsystem catalog

Description status:	DSSM V4.3B
Functional area:	Subsystem management
Domain:	SYSTEM-MANAGEMENT
Privileges:	SUBSYSTEM-MANAGEMENT

Function

Using this command, the system administration can extend the current subsystem configuration during a session (maximum number: 1000 subsystems and 16000 CALL entries). The catalog specified may either be a completely new one, which includes all the entries in the previous one, or it may contain only the new subsystems which are to be added into the current catalog.

In either case, the subsystem catalog specified must have been generated using SSCM. The ('old') subsystem catalog used during system initialization is not automatically updated. For the **next** session, the system administration can either

- use the catalog generated for ADD-SUBSYSTEM during system initialization, or
- generate a completely new and updated subsystem catalog and use this for system initialization. This new catalog need not build up quantitatively on an old predecessor catalog nor qualitatively support its references and attributes.

The ADD-SUBSYSTEM command is rejected if the class 5 memory reserved for the subsystem is insufficient (on system initialization, DSSM reserves class 5 memory for subsystems defined with MEMORY-CLASS=*LOCAL-PRIVILEGED, *LOCAL-UNPRIVILEGED or *BY-SLICE).

Format

ADD-SUBSYSTEM
CATALOG = <filename 1..54 without-gen-vers> ,TYPE = <u>*EXTENDED-ACTIVE-CONFIGURATION</u> / *NEW-SUBSYSTEMS

Operands

CATALOG = <filename 1..54 without-gen-vers>
Name of the new subsystem catalog.

TYPE =

Specifies whether the current catalog is to be extended or replaced.

TYPE = *EXTENDED-ACTIVE-CONFIGURATION

A completely new catalog is to be activated, containing not only the entries from its predecessor, but also the new subsystems.

TYPE = *NEW-SUBSYSTEMS

The specified catalog contains only new subsystems, which are to be added to the old catalog. DSSM will check the catalog to ensure that the subsystems which it contains really are new. If any subsystem is found which is also listed in the catalog which is being extended, the command will be rejected.

Restrictions

- The subsystems specified for generation using SSCM with RELATED-SUBSYSTEM and REFERENCED-SUBSYSTEM must be cycle-free, i.e. free of mutual dependency.
- It is not permitted to define different versions of a subsystem with the start attributes AT-SUBSYSTEM-CALL, BEFORE-SYSTEM-READY, AFTER-SYSTEM-READY, BEFORE-DSSM-LOAD, AT-DSSM-LOAD and MANDATORY-AT-STARTUP (exception: AT-SUBSYSTEM-CALL is permitted if coexistence is defined for all versions involved).

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
	32	ESM0296	Abnormal termination (REQM error)
	32	ESM0350	Internal DSSM problem during processing
	64	ESM0260	File not found
	64	ESM0261	Storage space limit reached in DSSM catalog
	64	ESM0262	File is not a catalog
	64	ESM0322	Maximum number of relations exceeded
	64	ESM0325	Inconsistent with old catalog
	64	ESM0332	Incompatible version of catalog
	64	ESM0340	Reserved address-space exhausted for subsystems with MEMORY-CLASS=*BY-SLICE

Notes

There is a problem with relations which refer to subsystems that are not defined in the current catalog (put there using the SSCM statement SAVE-CATALOG FORCED=*YES). This problem can be circumvented during a session by using SSCM to define the relation in a new catalog and then using ADD-SUBSYSTEM to define this new catalog.

*Example for TYPE=*EXTENDED-ACTIVE-CONFIGURATION*

Old catalog	→	New catalog
<pre>//START-CATALOG-CREATION old-cat //SET-SUBSYSTEM-ATTRIBUTES - // SUBSYSTEM-NAME=ss1, - // RELATED-SUBSYSTEM=ss2 //SAVE-CATALOG FORCED= - // *FOR-ADD-SUBSYSTEM</pre>		<pre>//START-CATALOG-CREATION new-cat //SET-SUBSYSTEM-ATTRIBUTES - // SUBSYSTEM-NAME=ss1, - // RELATED-SUBSYSTEM=ss2 //SET-SUBSYSTEM-ATTRIBUTES - // SUBSYSTEM-NAME=ss2 //SAVE-CATALOG</pre>

*Restrictions on the operand TYPE=*EXTENDED-ACTIVE-CONFIGURATION*

- Subsystems with the attribute MEMORY-CLASS=*LOCAL-PRIVILEGED, which are being added to the newly created catalog, must not exceed the size of the address space strip in user or system address space, nor may their location in the address space overlap with subsystems from the old catalog.
- The CREATION-TIME operand for any new subsystem must be chosen to be compatible with versions of the same subsystem which are already defined in the old catalog. In making this choice, the values BEFORE-SYSTEM-READY, AFTER-SYSTEM-READY, BEFORE-DSSM-LOAD, AT-DSSM-LOAD and MANDATORY-AT-STARTUP could be used, but would have no effect since the system startup time for a session which has started will already have been passed; i.e. the system administration will be given an appropriate warning, but the subsystem will not be loaded.
- In distributing subsystems to holder tasks (ASSIGN-HOLDER-TASK statement), the “stand alone” principle must be observed, i.e. subsystems from different catalogs may not be assigned to the same holder task.

Example

Subsystems in the old catalog: A, B, C
 Subsystems in the new catalog: A, B, C, D, E

Then:

```
//ASSIGN-HOLDER-TASK *SHARE-HOLDER(BY-SUB=(A,B)) is permissible
//ASSIGN-HOLDER-TASK *SHARE-HOLDER(BY-SUB=(D,E)) is permissible
```

but:

```
//ASSIGN-HOLDER-TASK *SHARE-HOLDER(BY-SUB=(A,D,C)) not permissible
```

- The new catalog must be larger than its predecessor, because it not only contains the old subsystems with their attributes (relations, dependencies, loading instructions), but must also maintain details of the new subsystems.

- Linkage and dependence relations (REFERENCED-SUBSYSTEM / RELATED-SUBSYSTEM) must not go outside the bounds of the catalog: for a subsystem defined in catalog A there must be no relations in that catalog which refer to a subsystem defined in catalog B.
- Once REMOVE-SUBSYSTEM has been used to delete a subsystem from the catalog, TYPE=*EXTENDED-ACTIVE-CONFIGURATION can no longer be specified.

It is not permissible to define different

- subsystems with an identical combination of the attributes: SVC-NUMBER / FUNCTION-NUMBER / FUNCTION-VERSION.
- subsystems with an identical combination of the attributes: FUNCTION-NUMBER / FUNCTION-VERSION (if the value *ALLOWED is set for VERSION-COEXISTENCE or VERSION-EXCHANGE) for subsystems which are indirectly linked via System Procedure Linkage (ISL).
- versions of a subsystem with an identical combination of the attributes: SVC-NUMBER / FUNCTION-NUMBER / FUNCTION-VERSION if the value *ALLOWED is set for VERSION-COEXISTENCE or VERSION-EXCHANGE.
- versions of a subsystem with an identical combination of the attributes: FUNCTION-NUMBER / FUNCTION-VERSION / VERSION-COEXISTENCE or VERSION-EXCHANGE for subsystems which are indirectly linked via System Procedure Linkage (ISL).

Overlaps of subsystems must be avoided. To prevent this, DSSM compares the values of the SIZE and START-ADDRESS operands in the SET-SUBSYSTEM-ATTRIBUTES statements.

The assignment of a holder task (by an SSCM statement) for an “old” and a new subsystem has no effect on holder task distribution.

Table of incompatibilities for relations between subsystems in the old and new catalog:

- x: this combination is not possible: neither link relations (REFERENCED-SUBSYSTEM) nor any other dependencies (RELATED-SUBSYSTEM) are permitted
- r: link relations (REFERENCED-SUBSYSTEM) are not permitted

Subsystem in the new catalog	Subsystem in the old catalog												
	MANDATORY-AT-STARTUP	BEFORE-SYSTEM-READY	AFTER-SYSTEM-READY	AT-CREATION-REQUEST	AT-SUBSYSTEM-CALL	BEFORE-DSSM-LOAD	AT-DSSM-LOAD	STOP-AT-SHUTDOWN=*YES	SUBSYSTEM-ACCESS=*LOW	SUBSYSTEM-ACCESS=*HIGH	MEMORY-CLASS=*LOCAL-PRIVILEGED	MEMORY-CLASS=*LOCAL-UNPRIVILEGED	MEMORY-CLASS=*BY-SLICE
MANDATORY-AT-STARTUP		x	x	x	x								
BEFORE-SYSTEM-READY			x	x	x								
AFTER-SYSTEM-READY				x	x								
BEFORE-DSSM-LOAD	x	x	x	x	x		x						
AT-DSSM-LOAD	x	x	x	x	x								
STOP-AT-SHUTDOWN=*NO								r					
SUBSYSTEM-ACCESS=*SYSTEM									x	x			x
MEMORY-CLASS=*SYSTEM-GLOBAL											x	x	
MEMORY-CLASS=*BY-SLICE											r	r	r

Table 27: Incompatibilities in relations between subsystems

ADD-USER

Create entry in user catalog

Description status:	SRPMNUC V19.0A
Functional area:	User management
Domain:	USER-ADMINISTRATION
Privileges:	STD-PROCESSING USER-ADMINISTRATION
Routing code:	\$

Function

The ADD-USER command enables systems support personnel to create an entry in the user catalog for a pubset, which means that a user catalog entry is set up for a user ID. If an entry is created for the user catalog of the *home pubset*, the access rights to the system (e.g. user ID, accounting number, user ID password, etc.) and the assignment of a default pubset for the user must be defined. Only in the user catalog of the home pubset is a check made whether the user has observed the correct syntax in the specifications (LOGON validation).

The system support staff enters all the necessary pubset-specific information in the user catalog of the *default pubset*. Systems support must define an upper limit for the user up to which the user may occupy storage space on this pubset. Additionally provisions can be made for the possibility of exceeding this limit.

The name of a new user ID can be chosen freely - within the set of permissible characters. The prefix 'SYS', however, is reserved for system IDs and must not be used.

The LOCK-USER=*YES operand can be used to set an access lock for the new user ID until all the desired attributes have been assigned and group membership has been declared. The entry must be stored on the home pubset; for data pubsets, the lock is irrelevant. The lock is retained until an UNLOCK-USER command is issued.

Restriction

The only nonprivileged users (STD-PROCESSING privilege) authorized to issue this command are those designated as group administrators. The actions a group administrator can take are defined by systems support. On setting up and managing user groups see the "SECOS" manual [35].

Format

```

ADD-USER

USER-IDENTIFICATION = <name 1..8>
, GROUP-IDENTIFICATION = *OWN / *UNIVERSAL / <name 1..8>
, PROTECTION-ATTRIBUTE = *PARAMETERS (...)
    *PARAMETERS(...)
        |
        | LOGON-PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16> / *SECRET
        | , PASSWORD-ENCRYPTION = *YES / *NO
        | , PASSWORD-MANAGEMENT = *BY-USER / *BY-ADMINISTRATOR / *USER-CHANGE-ONLY
        | , TAPE-ACCESS = *STD / *PRIVILEGED / *READ / *BYPASS-LABEL / *ALL
        | , FILE-AUDIT = *NO / *ALLOWED
, MAILING-ADDRESS = *NONE / <c-string 1..64 with-low> / <x-string 1..128>
, EMAIL-ADDRESS = *NONE / <composed-name 1..1800> / <c-string 1..1800 with-low> / <x-string 1..3600>
, PUBLIC-SPACE-LIMIT = *STD / *MAXIMUM / <integer 0..2147483647 2Kbyte>
, PUBLIC-SPACE-EXCESS = *NO / *ALLOWED / *TEMPORARILY-ALLOWED
, ADDRESS-SPACE-LIMIT = *STD / <integer 1..2147483647 Mbyte>
, MAX-ACCOUNT-RECORDS = 100 / *NO-LIMIT / <integer 0..32767>
, PROFILE-ID = *NONE / <structured-name 1..30>
, PUBSET = *HOME / <cat-id 1..4>
, RESIDENT-PAGES = *STD / *MAXIMUM / <integer 0..2147483647 4Kbyte>
, CSTMP-MACRO-ALLOWED = *NO / *YES
, DEFAULT-PUBSET = *HOME / <cat-id 1..4>
, TEST-OPTIONS = *PARAMETERS (...)
    *PARAMETERS(...)
        |
        | READ-PRIVILEGE = 1 / <integer 1..9>
        | , WRITE-PRIVILEGE = 1 / <integer 1..9>
        | , MODIFICATION = *CONTROLLED / *UNCONTROLLED
, AUDIT = *PARAMETERS (...)
    *PARAMETERS(...)
        |
        | HARDWARE-AUDIT = *ALLOWED / *NOT-ALLOWED
        | , LINKAGE-AUDIT = *ALLOWED / *NOT-ALLOWED
, DEFAULT-MSG-LANGUAGE = *STD / <name 1..1>
, FILE-NUMBER-LIMIT = *MAXIMUM / <integer 0..16777215>

```

(Part 1 of 2)

```

,JV-NUMBER-LIMIT = *MAXIMUM / <integer 0..16777215>
,TEMP-SPACE-LIMIT = *MAXIMUM / <integer 0..2147483647 2Kbyte>
,DMS-TUNING-RESOURCES = *NONE / *CONCURRENT-USE / *EXCLUSIVE-USE
,CODED-CHARACTER-SET = *STD / <name 1..8>
,PHYSICAL-ALLOCATION = *NOT-ALLOWED / *ALLOWED
,CRYPTO-SESSION-LIMIT = *STD / *MAXIMUM / <integer 0..32767>
,NET-STORAGE-USAGE = *ALLOWED / *NOT-ALLOWED
,ACCOUNT-ATTRIBUTES = *PARAMETERS (...)
  *PARAMETERS(...)
    ACCOUNT = <alphanum-name 1..8>
    ,CPU-LIMIT = *STD / *MAXIMUM / <integer 0..2147483647 seconds>
    ,SPOOLOUT-CLASS = 0 / *STD / <integer 0..255>
    ,MAXIMUM-RUN-PRIORITY = *STD / <integer 30..255>
    ,MAX-ALLOWED-CATEGORY = *STD / *TP / *SYSTEM
    ,PRIVILEGE = *NO / *PARAMETERS(...) / list-poss(3): *NO-CPU-LIMIT /
      *START-IMMEDIATE / *INHIBIT-DEACTIVATION
    *PARAMETERS(...)
      NO-CPU-LIMIT = *NO / *YES
      ,START-IMMEDIATE = *NO / *YES
      ,INHIBIT-DEACTIVATION = *NO / *YES
    ,POSIX-RLOGIN-DEFAULT = *NO / *YES
    ,LOGON-DEFAULT = *NO / *YES
,LOCK-USER = *NO / *YES

```

(Part 2 of 2)

Operands

USER-IDENTIFICATION = <name 1..8>

Defines the user ID to be entered in the catalog of the pubset in question.

GROUP-IDENTIFICATION =

Identifier of the user group to which the new user ID is to be assigned. SECOS can be used to install a hierarchy of user groups to each of which a number of user IDs can be assigned. Without this software product, all of the user IDs are subordinate to the group *UNIVERSAL, which acts as a dummy group in the system.

GROUP-IDENTIFICATION = *OWN

The user ID is assigned to the user group of the command issuer.

If the command issuer does not belong to a user group or SECOS is not installed, *UNIVERSAL is assumed.

GROUP-IDENTIFICATION = *UNIVERSAL

The user ID is explicitly assigned to the group *UNIVERSAL.

If SECOS is used, this value permits a user ID to be created outside of group hierarchies.

GROUP-IDENTIFICATION = <name 1..8>

Existing user group to which the new user ID is assigned.

This value can be specified only if the SECOS product is installed.

PROTECTION-ATTRIBUTE = *PARAMETERS(...)

Defines protection attributes.

LOGON-PASSWORD = *NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16> / *SECRET

Password protecting the user ID from unauthorized access (long password (<c-string 9..32>), see the MODIFY-USER-PROTECTION command).

PASSWORD-ENCRYPTION = *YES / *NO

The password of the user ID is encrypted after entry or is stored in its original form.

Password encryption presupposes that the system parameter ENCRYPT=Y is set in the parameter file (see the "Introduction to System Administration" [14]).

ADD-USER	parameter file	
	ENCRYPT=Y	ENCRYPT=N
PASS-ENCRYPT =*YES	encryption	No encryption
PASS-ENCRYPT =*NO	No encryption	No encryption

PASSWORD-MANAGEMENT =

Specifies the user's rights with regard to modification of his password.

PASSWORD-MANAGEMENT = *BY-USER

The user may define, modify or delete a password.

PASSWORD-MANAGEMENT = *BY-ADMINISTRATOR

Only systems support staff may define, modify or delete the password for the user ID.

PASSWORD-MANAGEMENT = *USER-CHANGE-ONLY

The user may define and modify a password. Deletion of the password, i.e. cancellation of access rights, is not permissible.

TAPE-ACCESS =

Defines whether error messages generated during label checking of tapes may be ignored by the operator for the user concerned.

TAPE-ACCESS = *STD

Error messages must not be ignored.

TAPE-ACCESS = *PRIVILEGED

The following error messages for input and output files may be ignored by the owner of the tape or by systems support staff:

- invalid VSN
- tape is write-protected
- incorrect file set identifier in the HDR1 label of the tape.

TAPE-ACCESS = *READ

The user may ignore error messages relating to input files; label checking is not deactivated. The following errors may result in messages during tape processing:

- invalid volume serial number (VSN)
- invalid file name
- invalid label on tape
- invalid access method
- invalid file sequence number on tape
- tape mark instead of end-of-volume label on tape
- double tape mark instead of end-of-volume label on tape.

TAPE-ACCESS = *BYPASS-LABEL

Label checking and thus any data protection for tape files is deactivated for tapes processed in INPUT or REVERSE mode. This privilege implies the TAPE-ACCESS=*READ function.

TAPE-ACCESS = *ALL

All error messages may be ignored. This privilege implies the TAPE-ACCESS=*PRIVILEGED and TAPE-ACCESS=*BYPASS-LABEL functions.

FILE-AUDIT = *NO / *ALLOWED

Defines whether the user is authorized to activate the AUDIT mode.

This mode serves to monitor DVS accesses to files or file generations by system exit routines or, if the SECOS software product is used, by the SAT component.

MAILING-ADDRESS = *NONE / <c-string 1..64 with-low> / <x-string 1..128>

Mailing address of the user for spoolout lists.

EMAIL-ADDRESS = *NONE / <composed-name 1..1800> / <c-string 1..1800 with-low> / <x-string 1..3600>

Specifies an email address or a list of email addresses for the users of this user ID. The addresses must be entered in the format '<local-part>@<domain>[, . . .]'. Optionally an address can also be prefixed by an address name in parentheses (see ADD-USER, "Example" on page 2-149). A list consists of multiple addresses separated by a comma and can only be specified as a string (c or x string).

The address or address list entered is evaluated when a file is sent by email (see the MAIL-FILE command).

PUBLIC-SPACE-LIMIT = *STD / *MAXIMUM / <integer 0..2147483647 2Kbyte>

Assigns the maximum storage space the user may occupy for his files on public volumes on the pubset allocated with the PUBSET operand. The specified value must be $\leq 2,147,483,647$.

PUBLIC-SPACE-LIMIT = *STD

By default the user is allocated 16,777,215 PAM pages.

PUBLIC-SPACE-LIMIT = *MAXIMUM

The user may use the full capacity of 2,147,483,647 PAM pages.

PUBLIC-SPACE-EXCESS =

Defines whether the user may exceed the limit defined by the PUBLIC-SPACE-LIMIT operand for the storage space on the allocated pubset.

PUBLIC-SPACE-EXCESS = *NO

The storage space limit must not be exceeded.

PUBLIC-SPACE-EXCESS = *TEMPORARILY-ALLOWED

In the case of storage space requests on the user's default pubset, the storage space limit may be exceeded, provided that it had not already been exceeded at logon time. For all other pubsets, PUBLIC-SPACE-EXCESS=*TEMPORARILY-ALLOWED has the same effect as PUBLIC-SPACE-EXCESS = *NO.

PUBLIC-SPACE-EXCESS = *ALLOWED

The storage space limit may be exceeded.

ADDRESS-SPACE-LIMIT = *STD / <integer 1..2147483647 Mbyte>

Defines how much user memory a task can request under this user ID. The user memory comprises both conventional class-6 memory (program space) and (data spaces). The maximum user memory depends on the size of the virtual memory and can be less than the value specified here.

ADDRESS-SPACE-LIMIT = *STD

The value of the system parameter SYSGJASL is assigned (the system parameter SYSGJASL has the default value 16 MB, see the “Introduction to System Administration” [14]).

MAX-ACCOUNT-RECORDS =

Defines how many user-specific accounting records for each job or program are allowed to be written into the accounting file of the system.

MAX-ACCOUNT-RECORDS = 100

The user is authorized to write 100 user-specific accounting records to the accounting file for each job or program. He is not allowed to write accounting records of his own.

MAX-ACCOUNT-RECORDS = *NO-LIMIT

The user is authorized to write any number of user-specific accounting records and accounting records of his own in the accounting file.

MAX-ACCOUNT-RECORDS = <integer 0..32767>

Defines how many user-specific accounting records can be written in the system accounting file for each job or program. The user is not authorized to add accounting records of his own.

PROFILE-ID =

Determines whether the user ID is assigned an SDF-PROFILE-ID. This PROFILE-ID characterizes a (SDF) group of user IDs which use a common group syntax file. Systems support can effect direct assignment of a user ID to a group syntax file by means of a suitable entry in the SDF parameter file.

PROFILE-ID = *NONE

The user ID is not assigned a profile ID and thus indirectly no group syntax file.

PROFILE-ID = <structured-name 1..30>

Defines the name of a profile ID which can be assigned to a group syntax file in the SDF parameter file.

PUBSET =

Defines the pubset whose user catalog is to accept the entry.

PUBSET = *HOME

The entry is made in the user catalog of the home pubset.

PUBSET = <cat-id 1..4>

Catalog identifier of the pubset whose user catalog is to accept the entry.

RESIDENT-PAGES = *STD / *MAXIMUM / <integer 0..2147483647 4Kbyte>

Defines the maximum number of resident main memory pages available to the user ID.

RESIDENT-PAGES = *STD

The user may use 32,767 resident main memory pages.

RESIDENT-PAGES = *MAXIMUM

The user may use 2,147,483,647 resident main memory pages.

CSTMP-MACRO-ALLOWED = *NO / *YES

Determines whether the user may use the CSTMP macro in his programs. The user can use the CSTMP macro to write-protect a memory pool (in class 6 memory) that can be shared by multiple users or explicitly cancel this protection. This feature is described in detail in the “Executive Macros” manual [22].

DEFAULT-PUBSET =

Assigns the user ID a default pubset on which the user can store his files and request storage space. If a file belonging to the newly entered user ID is accessed under any arbitrary user ID, and if the catalog ID is not specified, the catalog ID in the path name will automatically be extended by “.catid:”. Systems support can change the DEFAULT-PUBSET operand in any user catalog of an imported pubset. However, the user default pubset is determined only with the aid of the user catalog of the home pubset. For the TSOS user ID, the value of DEFAULT-PUBSET must be identical to the value of PUBSET.

DEFAULT-PUBSET = *HOME

Defines the home pubset to be the user default pubset.

DEFAULT-PUBSET = <cat-id 1..4>

Defines a pubset to be a user default pubset by means of the catalog ID.

TEST-OPTIONS = *PARAMETERS(...)

Defines the maximum possible privilege for testing and diagnostic analysis of program and hardware. The test privilege is interpreted by the software products AID and DAMP, by the access method ANITA and by hardware test and diagnostics products (TDPs) when performing maintenance work under the user ID which has the HARDWARE-MAINTENANCE system privilege (by default: SERVICE).

Defines the maximum permitted privilege levels for read and write access. Even at privilege level 2 the user has access to task-specific, sensitive data (system tables and control blocks). Higher values should be allocated only on request and for a limit period to selected user IDs. The values possible here and explanations are described in [section “Overview of test privileges” on page 1-106](#).

READ-PRIVILEGE = 1 / <integer 1..9>

Defines the maximum read privilege level.

WRITE-PRIVILEGE = 1 / <integer 1..9>

Defines the maximum write privilege level.

MODIFICATION =

Specifies whether the user requires the operator’s permission to modify his/her current test privilege.

MODIFICATION = *CONTROLLED

The user requires the operator’s permission.

MODIFICATION = *UNCONTROLLED

The user does not require the operator's permission.

AUDIT = *PARAMETERS (...)

Defines user-specific audit authorization. Authorization may be assigned to users separately for hardware audit and linkage audit. System-wide availability of the audit function is defined via the AUDALLOW system parameter.

Note Hardware audit is only supported on /390 architecture (SUs /390 and S servers).

HARDWARE-AUDIT = *ALLOWED / *NOT-ALLOWED

Specifies whether a user is authorized to control the hardware audit mode. The audit mode can be controlled by means of the START-, STOP-, HOLD- and RESUME-HARDWARE-AUDIT commands and via the AUDIT macro for the function states TU (task unprivileged) and TPR (task privileged). Modifications only affect the user ID's new tasks.

HARDWARE-AUDIT = *ALLOWED

The user is allowed to control the hardware audit mode, provided the audit function is available throughout the system.

HARDWARE-AUDIT = *NOT-ALLOWED

The user is not allowed to control the hardware audit mode.

LINKAGE-AUDIT = *ALLOWED / *NOT-ALLOWED

Specifies whether a user is authorized to control the linkage audit mode. The audit mode can be controlled by means of the START-, STOP-, HOLD- and RESUME-LINKAGE-AUDIT commands and via the AUDIT macro for the function states TU (task unprivileged), TPR (task privileged) and SIH (service interrupt handling). Modifications only affect the user ID's new tasks.

LINKAGE-AUDIT = *ALLOWED

The user is allowed to control the linkage audit mode, provided the audit function is available throughout the system.

LINKAGE-AUDIT = *NOT-ALLOWED

The user is not allowed to control the linkage audit mode.

DEFAULT-MSG-LANGUAGE = *STD / <name 1..1>

Specifies the language in which messages are output by default.

DEFAULT-MSG-LANGUAGE = *STD

The language defined using the MSGLPRI system parameter is used.

FILE-NUMBER-LIMIT =

Specifies the maximum number of files that may be created. This upper limit, or any lower value, may be assigned to subgroups or group members.

FILE-NUMBER-LIMIT = *MAXIMUM

The maximum number of files is 16777215.

FILE-NUMBER-LIMIT = <integer 0..16777215>

Specifies the maximum possible number of catalog entries as an exact number.

JV-NUMBER-LIMIT =

Specifies the maximum number of job variables that may be created. This upper limit, or any lower value, may be assigned to subgroups or group members.

JV-NUMBER-LIMIT = *MAXIMUM

The maximum number of job variables is 16777215.

JV-NUMBER-LIMIT = <integer 0..16777215>

Specifies the maximum possible number of job variables as an exact number.

TEMP-SPACE-LIMIT =

Specifies the maximum amount of temporary storage space which may be occupied on the shared volumes specified in the PUBSET operand.

TEMP-SPACE-LIMIT = *MAXIMUM

The maximum amount of storage space is 2147483647 PAM pages.

TEMP-SPACE-LIMIT = <integer 0..2147483647 2Kbyte>

Specifies exactly the amount of temporary storage space.

DMS-TUNING-RESOURCES =

Specifies which performance measures may be utilized, and the form in which they may be used.

DMS-TUNING-RESOURCES = *NONE

No tuning measures may be utilized.

DMS-TUNING-RESOURCES = *CONCURRENT-USE

The user may reserve preferred resources, but in doing so competes with all other users who have the same authorization.

DMS-TUNING-RESOURCES = *EXCLUSIVE-USE

The user may make exclusive reservations of preferred resources.

Permitted performance measures for the home and data pubsets

DMS-TUNING-RESOURCES=	PUBSET = *HOME			
	Resident ISAM pools	Resident FASTPAM environment	File attribute PERFORMANCE =*HIGH	=*VERY-HIGH
*NONE	no	no	no	-
*CONCURRENT-USE	yes	no	-	-
*EXCLUSIVE-USE	yes	yes	-	-

PUBSET = <Daten-Pubset>				
DMS-TUNING-RESOURCES=	Resident ISAM pools	Resident FASTPAM environment	File attribute PERFORMANCE	
			=*HIGH	=*VERY-HIGH
*NONE	-	-	no	no
*CONCURRENT-USE	-	-	yes	no
*EXCLUSIVE-USE	-	-	yes	yes

Home pubset	Data pubset	Permitted performance measures
*NONE	*NONE	– None
*CONCURRENT-USE	*NONE	– Resident ISAM pools
*EXCLUSIVE-USE	*NONE	– Resident ISAM pools – Resident FASTPAM environment
*NONE	*CONCURRENT-USE	– File attribute PERFORMANCE = HIGH on data pubset
*NONE	*EXCLUSIVE-USE	– File attribute PERFORMANCE = HIGH on data pubset – File attribute PERFORMANCE = VERY-HIGH on data pubset
*CONCURRENT-USE	*CONCURRENT-USE	– Resident ISAM pools
*CONCURRENT-USE	*EXCLUSIVE-USE	– Resident ISAM pools – File attribute PERFORMANCE = VERY-HIGH on data pubset
*EXCLUSIVE-USE	*CONCURRENT-USE	– Resident ISAM pools – Resident FASTPAM environment
*EXCLUSIVE-USE	*EXCLUSIVE-USE	– Resident ISAM pools – Resident FASTPAM environment – File attribute PERFORMANCE = VERY-HIGH on data pubset

Table 28: Permitted performance measures (ADD-USER command)

CODED-CHARACTER-SET = *STD / <name 1..8>

Specifies which CODED-CHARACTER-SET (CCS) is to be used. A name should only be specified here if the one required differs from the one preset by the system (*STD). A CCS identifies a character set (specific to a country) which is, for example, to be activated when files are being created. The specified CCS should be an EBCDIC character set.

PHYSICAL-ALLOCATION = *NOT-ALLOWED / *ALLOWED

Governs whether the user is allowed to perform physical space allocation (direct allocation) for the pubset.

PHYSICAL-ALLOCATION = *NOT-ALLOWED

The user is no longer allowed to perform physical space allocation for the pubset.

PHYSICAL-ALLOCATION = *ALLOWED

The user is now allowed to perform physical space allocation for the pubset.

CRYPTO-SESSION-LIMIT = *STD / *MAXIMUM / <integer 0..32767>

Defines the maximum number of openCRYPT sessions within a BS2000 session. The number of openCRYPT sessions already used is set to 0 at the start of a BS2000.

CRYPTO-SESSION-LIMIT = *STD

Defines the maximum number of 128 openCRYPT sessions.

CRYPTO-SESSION-LIMIT = *MAXIMUM

Defines the maximum number of 32767 openCRYPT sessions.

NET-STORAGE-USAGE = *ALLOWED / *NOT-ALLOWED

Specifies whether the user may occupy space on Net-Storage volumes.

NET-STORAGE-USAGE = *ALLOWED

The user is permitted to use Net-Storage volumes. The space occupied on the Net-Storage volume is not counted toward the user's PUBLIC-SPACE-LIMIT.

NET-STORAGE-USAGE = *NOT-ALLOWED

The user is not permitted to use Net-Storage volumes.

ACCOUNT-ATTRIBUTES = *PARAMETERS(...)

Specifications that are valid for only one account number of the user ID. The maximum number of account numbers per user ID is 60.

ACCOUNT = <alphanum-name 1..8>

Account number of the user ID, to which the following information refers.

CPU-LIMIT = *STD / *MAXIMUM / <integer 0..2147483647 seconds>

Total CPU time allocated to the user jobs under the specified account number.

CPU-LIMIT = *STD

The default value set in the SYSGJCPU system parameter (see the "Introduction to System Administration" [14]) is available.

CPU-LIMIT = *MAXIMUM

The time allotted the user is 2,147,483,647 CPU seconds.

SPOOLOUT-CLASS = 0 / <integer 0..255> / STD

Spoolout class for the first account number of the user ID

SPOOLOUT-CLASS = *STD

The default value set in the SYSGJCLA system parameter (see the "Introduction to System Administration" [14]) is available.

MAXIMUM-RUN-PRIORITY = *STD / <integer 30..255>

Specifies the highest priority that may be assigned jobs of the user ID (see note).

MAXIMUM-RUN-PRIORITY = *STD

The default value set in the SYSGJPRI system parameter (see the “Introduction to System Administration” [14]) [is available](#).

MAX-ALLOWED-CATEGORY =

This operand defines which task attributes user jobs are allowed to attain. If the user employs the TINF macro in his/her programs, a check is made in both the job class used by the job concerned and in the user catalog to see whether the right to use the task attribute TP was assigned to the user under the specified account number.

MAX-ALLOWED-CATEGORY = *STD

The tasks of the user can attain the task attributes BATCH and DIALOG.

MAX-ALLOWED-CATEGORY = *TP

The user jobs may attain the task attributes BATCH, DIALOG and TP.

MAX-ALLOWED-CATEGORY = *SYSTEM

All task attributes are permitted for the tasks of the user.

PRIVILEGE = *NO / *PARAMETERS(...) / list-poss(3): *NO-CPU-LIMIT / *START-IMMEDIATE / *INHIBIT-DEACTIVATION

Job management authorization declarations.

PRIVILEGE = *NO

The user ID is not given any job management privileges.

PRIVILEGE = *PARAMETERS(...)

A sequence of authorizations is to be agreed upon.

NO-CPU-LIMIT = *NO / *YES

Declares whether the user receives the authorization to run batch jobs without time restrictions.

NO-CPU-LIMIT = *YES

The user ID receives the authorization to run batch jobs without time restrictions under the specified account number. This applies even if the job class assigned to the job does not permit this start attribute. If the user specified the CPU-LIMIT=*NO operand in the SET-LOGON-PARAMETERS or ENTER-JOB command and this function is authorized neither in the user catalog nor in the job class assigned to the job, the batch job is rejected with an error message.

In the case of tasks without a time limit, the user’s CPU account is not debited.

START-IMMEDIATE = *NO / *YES

Defines whether the use is authorized to use the job express function.

START-IMMEDIATE = *YES

The user is authorized to use the job express function. With this authorization batch jobs are started immediately, even if the class limit of the job class in which the job concerned is to be started has been reached. This applies even if the definition of the job class does not permit this start attribute. If the EXPRESS function is not permitted either in the user catalog or in the job class definition, the batch job is accepted, but it is not started as an EXPRESS job.

INHIBIT-DEACTIVATION = *NO / *YES

Defines whether the user is authorized to inhibit deactivation.

INHIBIT-DEACTIVATION = *YES

The user ID is authorized to inhibit deactivation. The user's jobs are thus independent of the PRIOR function, by means of which jobs are placed in subordinate queues according to their system service requirements (macro time slice).

PRIVILEGE = *NO-CPU-LIMIT

The user is authorized to run batch jobs without time limitation under the specified account number.

PRIVILEGE = *START-IMMEDIATE

The user is authorized to use the job express function.

PRIVILEGE = *INHIBIT-DEACTIVATION

The user ID is authorized to inhibit deactivation.

POSIX-RLOGIN-DEFAULT = *NO / *YES

Defines whether the designated account number is to be used for POSIX remote login session accounting (the account number may likewise be used for BS2000 session accounting). If the user entry does not include an account number for POSIX remote login session accounting, access to the system by remote login is not possible. Systems support personnel working under the TSOS user ID have access even without an account number.

LOGON-DEFAULT = *NO / *YES

Defines whether the designated account number is to be used as the default account number for BS2000 timesharing mode if no account number is specified in the case of dialog or batch access.

LOCK-USER =

Defines whether the user ID is to be locked after its creation.

LOCK-USER = *NO

The user ID is not locked.

The user has free access to the system after entering his ID.

LOCK-USER = *YES

The user ID is locked.

The system support staff is thus able to assign all desired attributes to the user ID and block premature access by the user. If the product SECOS is used, the system support staff can achieve integration in the group structure and the protective mechanisms for the password in this way without thereby permitting SET-LOGON-PARAMETERS on the ID in question. The access lock must be explicitly canceled (UNLOCK-USER command) by the system support staff.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
2	0	SRM6001	Command executed with a warning
	1	SRM6010	Syntax error
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error
	130	SRM6030	Command cannot temporarily be executed

Notes

- Task scheduling priorities

The task scheduling priorities (30-255) for job control are defined:

- in the user catalog (ADD-USER command, MAXIMUM-RUN-PRIORITY operand)
- in the job class definition (JMU statement DEFINE-JOB-CLASS, RUN-PRIO operand; in addition to the default priority, a maximum priority can be defined)

If the user specifies a task scheduling priority in the SET-LOGON-PARAMETERS or ENTER-JOB command, then this priority is checked both in the user catalog and in the job class assigned to the job. An example is given below:

Priority in the SET-LOGON-PARAMETERS or ENTER-JOB command	Priority in the job class		Priority in the user catalog	Priority with which the job is started
	DEFAULT	MAXIMUM		
200	190	150	180	200
150	190	150	180	150
130	190	150	180	190
-	190	150	180	190
130	190	150	130	130
130	190	150	200	200
200	190	NO	180	200

Priority in the SET-LOGON-PARAMETERS or ENTER-JOB command	Priority in the job class		Priority in the user catalog	Priority with which the job is started
	DEFAULT	MAXIMUM		
170	190	NO	180	190
130	190	NO	180	190
-	190	NO	180	190

If the priority specified by the user in the SET-LOGON-PARAMETERS or ENTER-JOB command is **worse** than the best priority of the job class and in the user catalog (i.e. allowed either in the job class or in the user catalog), the job is started with the priority specified by the user. If the priority specified by the user in the SET-LOGON-PARAMETERS or ENTER-JOB command is **better** than the priorities of the job class and in the user catalog (i.e. not allowed either in the job class or in the user catalog), the job is given whichever is worse, the default priority of the job class or the priority in the user catalog.

If the user does not specify any priority in the SET-LOGON-PARAMETERS or ENTER-JOB command, the job is started with the default priority.

- Threshold values for files, job variables, temporary memory and address space

These threshold values are defined for the user ID using the FILE-NUMBER-LIMIT, JV-NUMBER-LIMIT, TEMP-SPACE-LIMIT and ADDRESS-SPACE-LIMIT operands. The user ID TSOS may exceed the defined limits. All other user IDs must always comply with the defined limits.

When creating or modifying a user ID, a global user administrator may specify values which exceed the group potential of these limit values.

Example

```
/add-user user-id=cognibs3,prot-attr=(logon-pass='8743ryz9',
                                     pass-manag=*user-cha-only,tape-access=*read),
mail-addr=c'Abteilung Z8 Raum 55.105',
email-addr=c'alfred.holli@incognito.de,joachim.vogi@incognito.de,
           (jk)johannes.kuli@incognito.de,(mr)mathias.reh@incognito.de',
pub-space-lim=20000,
account-attr=(account=acc00015,cpu-limit=10000,privil=*start-immed)
```

```
/show-user-attr cognibs3
%SHOW-USER-ATTRIBUTES --- PUBSET TK82 - USER COGNIBS3          2014-01-29 15:09:50
%-----
%USER-ID                COGNIBS3                PUBLIC-SPACE-USED          0
%GROUP-ID              *UNIVERSAL              PUBLIC-SPACE-LIMIT        20000
%DEFAULT-PUBSET        TK82                    PUBLIC-SPACE-EXCESS       *NO
%MAX-ACCOUNT-RECORDS  100                    TEMP-SPACE-USED          0
%DEFAULT-MSG-LANGUAGE                                TEMP-SPACE-LIMIT        2147483647
%                                                                FILES                    0
%PROTECTION-ATTRIBUTES...                               FILE-NUMBER-LIMIT       16777215
%LOGON-PASSWORD        *YES                    JOB-VARIABLES            0
%PASSWORD-MGMT        *USER-CHANGE-ONLY    JV-NUMBER-LIMIT         16777215
%TAPE-ACCESS          *READ                    RESIDENT-PAGES           32767
%FILE-AUDIT           *NO                    ADDRESS-SPACE-LIMIT      16
%                                                                DMS-TUNING-RESOURCES    *NONE
%TEST-OPTIONS...                                           CSTMP-MACRO-ALLOWED    *NO
%READ-PRIVILEGE        1                    CODED-CHARACTER-SET      EDF03IRV
%WRITE-PRIVILEGE       1                    PHYSICAL-ALLOCATION       *NO
%MODIFICATION          *CONTROLLED           USER-LOCKED             *NO
%                                                                CRYPTO-SESSION-USED    0
%AUDIT...                                                       CRYPTO-SESSION-LIMIT   128
%HARDWARE-AUDIT        *ALLOWED              NET-STORAGE-USAGE       *ALLOWED
%LINKAGE-AUDIT        *ALLOWED
%
%PROFILE-ID *NONE
%MAIL-ADDRESS Abteilung Z8 Raum 55.105
%EMAIL-ADDRESS alfred.holli@incognito.de,
%              joachim.vogi@incognito.de,
%              (jk)johannes.kuli@incognito.de,
%              (mr)mathias.reh@incognito.de
%
%+-----+-----+-----+-----+-----+-----+-----+-----+
%!ACCOUNT-#! CPU-LIMIT !SPOOLOUT-!MAX-RUN-!MAX-ALLOWED-!NO-CPU-!START-!INHIB-!
%!          !          ! CLASS !PRIORITY! CATEGORY ! LIMIT ! IMMED! DEACT!
%+-----+-----+-----+-----+-----+-----+-----+-----+
%! ACC00015!   10000!    0 !   255 !         STD ! NO ! YES ! NO !
%+-----+-----+-----+-----+-----+-----+-----+-----+
%DEFAULT-ACCOUNT-# FOR LOGON:          *NONE
%DEFAULT-ACCOUNT-# FOR REMOTE-LOGIN:  *NONE
%
%DEFAULT-JOB-CLASS FOR BATCH-JOBS:    JCBSTD
%DEFAULT-JOB-CLASS FOR DIALOG-JOBS:   JCDSTD
%LIST OF JOB-CLASSES ALLOWED:
%JCBATCHF JCBSTD  JCB00050 JCB00200 JCB02000 JCB05000 JCB32000 JCDSTD
%-----
%SHOW-USER-ATTRIBUTES                END OF DISPLAY FOR USER COGNIBS3 ON PUBSET TK82
/
```

AGOGO

Continue command file

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Editing command files
Domain:	not allocated
Privileges:	OPERATING
Routing code:	E

Function

The AGOGO command is used for processing a command file that has been placed in the wait state. Depending on the conditions for continuation, several AGOGO commands may be required, see the ASTOP command.

This command is intended for use in authorized user programs which need to be synchronized with the processing of a command file.

The operator should use this command only in the following situation:

If an authorized user program has been started from a command file and an error has occurred during its execution, the operator can use an AGOGO command (or several, if required) to shorten the wait time which would otherwise elapse before processing is continued. The wait time is 3 minutes, unless a different value has been set using the class 2 system parameter NBRUNWT.

The command must not be issued from a user task with the operating privilege.

The way this command works is described under “Command files for the operator” in the “Introduction to System Administration” [14].

Restrictions when using the “Operator LOGON” function

If the “Operator LOGON” function is used (incompatible mode; system parameter NBCONOPI=Y), the AGOGO command is not accepted after “SYSTEM READY” unless the originating task is running under TSOS or under the user ID which entered the RUN command. The AGOGO command is also allowed if the RUN or AGOGO command was issued by an authorized user program with generated authorization names or if the command issuer has been disconnected. In all other cases, the AGOGO command is rejected with message NBR1013.

This restriction does not apply to the CMDFILE running at “SYSTEM READY” time or to any command sequences started from it.

Format

AGOGO

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
1	0	EXC0916	No command file in ASTOP state
2	0	NBR1008	Command acknowledged. Another AGOGO command expected
	64	NBR1013	No authorization

ASR

Assign routing code

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Operator terminal control
Domain:	not allocated
Privileges:	OPERATING
Routing code:	E

Function

The ASR command enables the operator to carry out the following functions:

- Assign routing codes to operator terminals (consoles) or authorized user programs with generated authorization names.
Messages with these routing codes are sent to the appropriate operator terminals or authorized user programs. Commands with these routing codes may be issued from the appropriate operator terminals or authorized user programs.
- Assign routing codes to operator terminals (consoles) or authorized user programs with generated authorization names.
- Delete the assigned routing codes for operator terminals or authorized user programs with generated authorization names.
- Output information on the assignment of routing codes and filter levels to operator terminals or authorized user programs with generated authorization names.



Authorized user programs with dynamic authorization names can use ASR only to query their own set of routing codes. Assignment and deletion is done with the CREATE-OPERATOR-ROLE and DELETE-OPERATOR-ROLE commands.

Restrictions

The command is available in ISP format only; in other words, there are no SDF functions (such as syntax analysis or help). The format has been subdivided into four subformats:

- Format 1: Output information, starting on [page 2-153](#)
- Format 2: Assign routing codes starting on [page 2-153](#)
- Format 3: Cancel assignment starting on [page 2-155](#)
- Format 4: Reset assignment to time of system startup starting on [page 2-157](#)

The meaning of system parameters ASRSW1 and ASRSW2 depends on whether the “Operator LOGON” function is used (system parameter NBCONOPI=Y / N); see Note [on page 2-160](#).

Format 1

Operation	Operands
ASR	$\left\{ \begin{array}{l} \text{HELP} \\ \text{H} \end{array} \right\} [, \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{CONSOLE} \\ \text{CS} \end{array} \right\} = \left\{ \begin{array}{l} \text{bn} \\ (\text{bn1}, \dots) \\ \text{mn} \\ (\text{mn1}, \dots) \\ \text{ALL} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{CODE} \\ \text{CD} \end{array} \right\} = \left\{ \begin{array}{l} \text{bs} \\ (\text{bs1}, \dots) \\ \text{ALL} \end{array} \right\} \end{array} \right\}]$

Operands

HELP

Gives information on the assignment of routing codes to operator terminals or authorized user programs.

without operand

Gives information on the authorization name of the authorized user program or on the mnemonic name of the operator terminal from which the ASR command was issued.

CONSOLE =up =(up1,...) Outputs the following information for the user programs “up” (“up” is the 4-character authorization name of a user program):

- CODE=..., shows the routing codes assigned
- INOP, there is no connection with \$CONSOLE
- PROCESSOR=..., the name of the processor from which the connection was set up
- STATION=..., the name of the station of the authorized user program from which the connection was set up

=mn =(mn1,...) Outputs the following information for the operator terminals “mn” (“mn” is the mnemonic device name of an operator terminal):

- CODE=..., shows the routing codes assigned
- INOP, operator terminals “mn” not operable
- NOINF, receipt of information messages is suppressed
- SWITCHED OFF, an operator terminal has been switched over to the standby operator terminal; see the CONSOLE command

=ALL Information as above is output for all operator terminals and all authorized user programs. CONSOLE=ALL and CODE=ALL are mutually exclusive.

Note

Routing codes which are temporarily assigned to a standby operator terminal because of a main operator terminal failure are also output. If the “Operator LOGON” function is used (system parameter NBCONOPI=Y), the operator terminal “OFF” condition is no longer indicated. The “INACTIVE” condition is indicated for physical operator terminals at which no operator has logged on.

CODE =rc =(rc1,...) Lists the operator terminals and authorized user programs to which the routing codes “rc” are assigned.

=ALL Lists information on all routing codes defined in the system. CONSOLE=ALL and CODE=ALL are mutually exclusive.

Note

The routing codes which are temporarily assigned to a standby operator terminal because of a main operator terminal failure are not output.

Format 2

Operation	Operanden
ASR	$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{CODE} \\ \text{CD} \end{array} \right\} = \left\{ \begin{array}{l} \text{bs} \\ (\text{bs1}, \dots) \\ \text{ALL} \end{array} \right\} \\ \\ \left\{ \begin{array}{l} \text{CONSOLE} \\ \text{CS} \end{array} \right\} = \left\{ \begin{array}{l} \text{bn} \\ (\text{bn1}, \dots) \\ \text{mn} \\ (\text{mn1}, \dots) \\ \text{ALL} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{CODE} \\ \text{CD} \end{array} \right\} = \left\{ \begin{array}{l} \text{bs} \\ (\text{bs1}, \dots) \\ \text{ALL} \end{array} \right\} \right] \end{array} \right\} \left\{ \begin{array}{l} \text{ADD} \\ \text{A} \end{array} \right\}$

Operands

ADD Assigns a set of routing codes to operator terminals or authorized user programs with generated authorization names.



This function must neither be entered by user programs with dynamic authorization names themselves nor used with reference to them. In the mode with operator LOGON (system parameter NBCONOPI=Y), it must also not be used on physical operator terminals (including Teleservice).

CODE =rc =(rc1,...) The routing codes “rc” are assigned to the operator terminal or authorized user program from which the ASR command was issued.

=ALL All routing codes defined in the system are assigned to the operator terminal or authorized user program from which the ASR command was issued.

CONSOLE =up =(up1,...) For main operator terminal only: The main operator terminal assigns itself all routing codes assigned to the user programs with the generated authorization name “up”.

=mn =(mn1,...) The main operator terminal assigns itself all routing codes assigned to the operator terminals “mn”.

=ALL

The main operator terminal assigns itself all routing codes assigned to other operator terminals or user programs with generated authorization names.

Note

The routing code assignment for the operator subterminals and user programs remains intact. The routing codes previously assigned to the main operator terminal are also retained. See also Note [on page 2-160](#).

CONSOLE=...,CODE=... For main operator terminal only: The main operator terminal assigns the routing codes "rc" (specified in the CODE operand) to the operator terminals "mr" or the user programs "up" (specified in the CONSOLE operand). See also Note [on page 2-160](#).

Format 3

Operation	Operanden
ASR	$\left\{ \begin{array}{l} \text{DELETE} \\ \text{D} \end{array} \right\}, \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{CODE} \\ \text{CD} \end{array} \right\} = \left\{ \begin{array}{l} \text{bs} \\ (\text{bs1}, \dots) \\ \text{ALL} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{CONSOLE} \\ \text{CS} \end{array} \right\} = \left\{ \begin{array}{l} \text{bn} \\ (\text{bn1}, \dots) \\ \text{mn} \\ (\text{mn1}, \dots) \\ \text{ALL} \end{array} \right\}, \left\{ \begin{array}{l} \text{CODE} \\ \text{CD} \end{array} \right\} = \left\{ \begin{array}{l} \text{bs} \\ (\text{bs1}, \dots) \\ \text{ALL} \end{array} \right\} \end{array} \right\}$

Operands

DELETE Cancels the assignment of routing codes to operator terminals or authorized user programs with generated authorization names.



This function must neither be entered by user programs with dynamic authorization names themselves nor used with reference to them. In the mode with operator LOGON (system parameter NBCONOPI=Y), it must also not be used on physical operator terminals (including Teleservice).

CODE =rc =(rc1,...) The assignment of the routing codes "rc" to the operator terminal or authorized user program from which the ASR command was issued is canceled.

=ALL The assignment of all routing codes to the operator terminal or authorized user program from which the ASR command was issued is canceled.

CONSOLE=...,CODE=... For main operator terminal only: The main operator terminal cancels the assignment of the routing codes specified in the CODE operand to the operator terminals or authorized user programs specified in the CONSOLE operand. See also Note [on page 2-160](#).

Format 4

Operation	Operanden
ASR	$ \left. \begin{array}{l} \left\{ \begin{array}{l} \text{CODE} \\ \text{CD} \end{array} \right\} = \left\{ \begin{array}{l} \text{bs} \\ (\text{bs1}, \dots) \\ \text{ALL} \end{array} \right\} \\ \\ \left. \left[\begin{array}{l} \text{CONSOLE} \\ \text{CS} \end{array} \right\} = \left\{ \begin{array}{l} \text{bn} \\ (\text{bn1}, \dots) \\ \text{mn} \\ (\text{mn1}, \dots) \\ \text{ALL} \end{array} \right\} \right\} \left\{ \begin{array}{l} \text{PRIMARY} \\ \text{P} \end{array} \right\} \end{array} \right. $

Operands

PRIMARY Resets the assignment of routing codes to the state which existed at system startup.



This function must neither be entered by user programs with dynamic authorization names themselves nor used with reference to them. In the mode with operator LOGON (system parameter NBCONOPI=Y), it may only be used in authorized user programs with generated authorization names and is also allowed only with reference to them. Other restrictions depend on system parameters ASRSW1 and ASRSW2 (see Note [on page 2-160](#))

without operand The assignment of all routing codes to the operator terminal or authorized user program which issued the command is reset to the state which existed at system startup.

CODE

=rc =(rc1,...) For the main operator terminal only: The main operator terminal resets the assignment of the routing codes “rc” to the state which existed at system startup.

=ALL The main operator terminal resets the assignment of all routing codes defined in the system to the state which existed at system startup. See also Note [on page 2-160](#).

CONSOLE =up =(up1,...) For main operator terminal only: The main operator terminal resets the assignment of all routing codes to the user programs “up” to the state which existed at system startup.

=mn =(mn1,...) The main operator terminal resets the assignment of all routing codes to the operator terminals “mn” to the state which existed at system startup.

=ALL The main operator terminal resets the assignment of all routing codes defined in the system to the state which existed at system startup. See also Note [on page 2-160](#).

Return codes

The following functions of the ASR command, referred to in the “Meaning” column below, were replaced by new commands in BS2000/OSD-BC V2.0 and are not included in the present command description:

ASR function	Replaced by
ASR MAIN	/REQUEST-MAIN-CONSOLE-FUNCTIONS
ASR DESTINATION	/SHOW-SYSTEM-PARAMETERS
ASR SUPPRESS,<tsn>	/RESET-MSG-BUFFER

(SC2)	SC1	Maincode	Meaning
0	0	CMD0001	Command executed successfully
1	0	NBR0823	This is the main operator terminal. “ASR MAIN” and /REQUEST-MAIN-CONSOLE-FUNCTIONS ignored
2	0	NBR1031	Command only partially executed because it includes invalid actions relating to authorized user programs with dynamic authorization names
2	0	NBR1036	This operand combination can only be partially processed because NBCONOPI=Y is set
0	1	CMD0202	Syntax error
0	2	CMD0198	Shutdown
0	2	NBR0724	“ASR SUPPRESS” is not supported with NBCONOPI=Y. Use /RESET-MSG-BUFFER
0	2	NBR1033	“ASR MAIN” is not supported with NBCONOPI=Y. Use /REQUEST-MAIN-CONSOLE-FUNCTIONS
0	2	NBR1034	“ASR DESTINATION” is not supported with NBCONOPI=Y. Use /SHOW-SYSTEM-PARAMETERS
0	64	EXC0053	The entered operand combination is valid for the main operator terminal only
0	64	NBR0722	“ASR SUPPRESS” and /RESET-MSG-BUFFER allowed only on physical operator terminals
0	64	NBR0821	“ASR MAIN” and /REQUEST-MAIN-CONSOLE-FUNCTIONS rejected by main operator terminal
0	64	NBR0822	“ASR MAIN” and /REQUEST-MAIN-CONSOLE-FUNCTIONS allowed only on physical operator terminals
0	64	NBR0898	Command allowed only on operator terminals
0	64	NBR1030	Routing code assignments may not be modified with the ASR command either for or by authorized user programs with dynamic authorization names

(Part 1 of 2)

(SC2)	SC1	Maincode	Meaning
0	64	NBR1032	Filters for authorized user programs with dynamic authorization names may only be modified by those programs
0	64	NBR1035	This operand combination is not allowed on this operator terminal because NBCONOPI=Y is set
0	128	EXC0056	"ASR MAIN" and /REQUEST-MAIN-CONSOLE-FUNCTIONS in use. Try again later
0	130	EXC0054	Space request error
0	130	EXC0057	Task saturation

(Part 2 of 2)

Notes

- The following maximum values may be specified in the various operands of an ASR command:
 - in the operand CODE=(rc1,...) 12 routing codes,
 - in the operand CONSOLE=(mn1,...) 24 operator terminals,
 - in the operand CONSOLE=(up1,...) 38 authorized user programs.
- If a keyword operand contains a mnemonic device name of an operator terminal or an authorization name of a user program which is not defined in the system, one of the following messages appears:

```
CONSOLE 'bn' NOT FOUND
```


3. If the ASR command contains a number of entries which are not defined in the system, a separate message is output for each invalid entry.

Example: `CONSOLE=(K1,B3)` are not defined in the system, `CONSOLE=(XY,KL)` are defined. The command is therefore given as follows:

```
/ASR A,CD=(A,B,C,X,Y,Z),CS=(K1,B3,XY,KL)
```

The following messages appear as a response:

```
CONSOLE K1 NOT FOUND
CONSOLE B3 NOT FOUND
NBR0740  COMMAND COMPLETED 'ASR'; RESULT: SC2=0, SC1=0, MC=CMD0001)
```

The command is still executed, but only the correct entries in the operands are taken into account. In other words, the command in the example is equivalent to

```
/ASR A,CD=(A,B,C,X,Y,Z),CS=(XY,KL)
```

4. Whether input from operator subterminals is admissible is specified with the system parameter `ASRSW1` (see the “Introduction to System Administration” [14]). If input from operator subterminals is permitted, the operands are executed in so far as they affect subterminal status.
5. It is possible to influence the admissibility of an input from an authorized user program with generated authorization name by means of the class 2 system parameters `ASRSW1` and `ASRSW2` (see the “Introduction to System Administration” [14]).
6. The ASR command can always be input from the main operator terminal in the mode without operator LOGON. If the “Operator LOGON” function is used (system parameter `NBCONOPI=Y`), the main operator terminal must have the required routing code.
7. The meaning of system parameters `ASRSW1` and `ASRSW2` depends on whether the “Operator LOGON” function is used (system parameter `NBCONOPI=Y / N`). The effects of the various combinations are shown in Tables [table 29](#) and [30 on page 2-162](#). The following considerations (not included in the tables) are generally applicable:
 - Authorized user programs with dynamic authorization names are unaffected by the settings of the system parameters, because they have no influence on the distribution of routing codes and because their own routing codes cannot be modified from other operator terminals.

- In the mode with operator LOGON (NBCONOP1=Y), regardless of the settings of the system parameters, the main operator terminal is allowed to modify the routing codes of authorized user programs with generated authorization names, provided that it has input authorization. In the mode without operator LOGON (NBCONOP1=N) it always has input authorization and is further allowed to modify the routing codes of all physical operator terminals.

Mode without operator LOGON: NBCONOP1 = N		
ASRSW1	ASRSW2	Effect
0	0	Authorized user programs with generated authorization names are allowed to modify their own routing codes
0	1	Physical operator subterminals are allowed to modify their own routing codes. Authorized user programs with generated authorization names are allowed to modify their own routing codes.
1	0	Only the main operator terminal is allowed to modify routing codes, and it may do so only for authorized user programs with generated authorization names
1	1	Authorized user programs are allowed to modify each other's routing codes

Table 29: Effect of system parameters ASRSW1 and ASRSW2 (mode with operator LOGON)

Mode with operator LOGON: NBCONOP1 = Y		
ASRSW1	ASRSW2	Effect
0	0	Physical operator subterminals and authorized user programs with generated authorization names are allowed to modify their own routing codes
0	1	Only the main operator terminal and authorized user programs with generated authorization names are allowed to modify routing codes, but they may do so for all operator terminals
1	0	Only the main operator terminal is allowed to modify routing code assignments, but it may do so for all operator terminals
1	1	Only the main operator terminal and authorized user programs with generated authorization names are allowed to modify routing codes, but they may do so for all operator terminals

Table 30: Effect of system parameters ASRSW1 and ASRSW2 (mode with operator LOGON)

ASSIGN-STREAM

Assign S variable stream

Description status:	SDF-P-BASYS V2.5E
Functional area:	Procedures
Domain:	PROCEDURE
Privileges:	STD-PROCESSING OPERATING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

The ASSIGN-STREAM command assigns an S variable stream to a variable or a server (e.g. FHS). See also “S variable streams” in the “SDF-P” manual [34].

Restrictions

Assignment of an S variable stream to a variable using TO=*VARIABLE(...) is not possible unless the chargeable SDF-P subsystem is loaded.

Format

```

ASSIGN-STREAM

STREAM-NAME = SYSVAR / SYSMSG / SYSINF / <structured-name 1..20>
, TO = *STD / <structured-name 1..20> / *DUMMY / *SAME-AS-CALLING-PROC / *VARIABLE(...) /
*SERVER(...)
*VARIABLE(...)
| VARIABLE-NAME = *NONE / <composed-name 1..255>(…)
| <composed-name>(…)
| | WRITE-MODE = *EXTEND / *PREFIX
, RETURN-VARIABLE-NAME = *NONE / <composed-name 1..255>(…)
| <composed-name 1..255>(…)
| | WRITE-MODE = *EXTEND / *PREFIX
, CONTROL-VAR-NAME = *NONE / <composed-name 1..255>(…)
| <composed-name 1..255>(…)
| | WRITE-MODE = *EXTEND / *PREFIX
, RET-CONTROL-VAR-NAME = *NONE / <composed-name 1..255>(…)
| <composed-name 1..255>(…)
| | WRITE-MODE = *EXTEND / *PREFIX
*SERVER(...)
| SERVER-NAME = <structured-name 1..30>
, SERVER-INFORMATION = *NONE / <c-string 1..1800>
    
```

Operands

STREAM-NAME = <structured-name 1..20> / **SYSVAR** / **SYSMSG** / **SYSINF**

Assigns an S variable stream. The operand values **SYSINF**, **SYSMSG** and **SYSVAR** are reserved words. They must not be abbreviated.

SYSINF: contains structured outputs from commands and programs

SYSMSG: contains structured guaranteed messages

SYSVAR: contains information from **SYSINF** and **SYSVAR**; however, it is still possible to process the different data items separately.

TO =

Specifies the server which is assigned to the current S variable stream.

TO = *STD

Default assignment. The following table indicates the default values used internally by TO for the various combinations featuring the STREAM-NAME operand.

STREAM-NAME=	TO=*STD	Information contained
SYSINF	SYSVAR	structured outputs from commands and programs
SYSMSG	SYSVAR	Structured guaranteed messages
SYSVAR	*DUMMY	structured outputs from commands and programs and structured guaranteed messages
<structured-name 1..20>	*DUMMY	User variable stream

TO = <structured-name 1..20>

Name of the user server. Any loops in chains of S variable stream assignments will be rejected; e.g.

```

ASSIGN-STREAM S3,*DUMMY
ASSIGN-STREAM S2,S3
ASSIGN-STREAM S3,S2          -----> SDP0511
    
```

TO = *DUMMY

No assignment. Transmitted variables are lost. The client is informed of this by a warning.

TO = *SAME-AS-CALLING-PROC

Assigns the calling procedure's server. If there is no assignment in the calling procedure, the assignment is rejected and the S variable stream remains unaltered.

TO = *VARIABLE(...)

The server is SDF-P. The transmitted variables are written to the specified S variables, or read from it.

Specification is only possible if the chargeable SDF-P subsystem is loaded. The operands of the *VARIABLE(...) structure are described in full in the "SDF-P" manual [34].

TO = *SERVER(...)

Links the S variable stream with the specified server.

SERVER-NAME = <structured-name 1..30>

Name of the server.

SERVER-INFORMATION =

Information which must be sent to the server, e.g. the name of the format library for FHS.

SERVER-INFORMATION = *NONE

No information must be sent to the server.

SERVER-INFORMATION = <c-string 1..1800>

Text of the message, in the form of a string.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
2	0	SDP0531	Warning by the server; process being continued
	1	CMD0202	Syntax error
	3	CMD2203	Incorrect syntax file
	32	CMD0221	System error (internal error)
	64	CMD0216	User does not have the required privilege
	64	SDP0091	Semantic error
	64	SDP0532	Server error; command rejected
	64	SDP0534	Internal server error; command terminated. Server link terminated following unexpected event or due to shortage or absence of system resources
	130	SDP0099	No further address space available

Example

See the SHOW-STREAM-ASSIGNMENT and TRANSMIT-BY-STREAM commands in the "SDF-P" manual [34].

ASSIGN-SYSDTA

Assign SYSDTA to input source

Description status:	SYSFILE V19.0A
Functional area:	Job processing File processing
Domain:	JOB PROCEDURE PROGRAM
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

The ASSIGN-SYSDTA command assigns the system (input) file SYSDTA to an input source. The primary assignment and attributes of SYSDTA are described in [section "System files" on page 1-73](#).

Restrictions

Allocating a system file to a variable with TO=*VARIABLE(...) is only possible if the chargeable subsystem SDF-P is loaded.

Format

ASSIGN-SYSDTA
<p>TO = <filename 1..54> / *VARIABLE(...) / *LIBRARY-ELEMENT(...) / *PRIMARY / *SYSCMD / *DISKETTE(...)</p> <p>*VARIABLE(...)</p> <p> VARIABLE-NAME = <composed-name 1..254></p> <p>*LIBRARY-ELEMENT(...)</p> <p> LIBRARY = <filename 1..54 without-gen-vers></p> <p> ELEMENT = <composed-name 1..64 with-under>(...</p> <p> <composed-name 1..64 with-under>(...</p> <p> VERSION = *HIGHEST-EXISTING / *STD / <composed-name 1..24></p> <p> TYPE = <u>S</u> / <alphanum-name 1..8></p> <p>DATA-ESCAPE-CHAR = *COMPATIBLE / '&&' / '#' / '*' / '@' / '\$' / *STD</p>

Operands

TO =

Input source to which SYSDTA is to be assigned.

TO = <filename 1..54>

Name of the file to which SYSDTA is to be assigned. The file must be cataloged with the following attributes:

- variable-length records
- SAM or ISAM access method
- start of ISAM key: byte 5
- length of ISAM key: 8 bytes

TO = *VARIABLE(...)

Name of a complex S variable of the “list” type (i.e. a series of elements of the same variable type which can only be accessed sequentially). Complex S variables can only be used if the chargeable subsystem SDF-P is loaded (see the “SDF-P” manual [34]). The command will be rejected if the specified list has not been declared or if the list elements are not of variable type STRING (or ANY).

VARIABLE-NAME = <composed-name 1..254>

Name of the S variable.

TO = *LIBRARY-ELEMENT(...)

The input source is a PLAM library element.

LIBRARY = <filename 1..54 without-gen-vers>

Name of a PLAM library.

ELEMENT = <composed-name 1..64 with-under>(…)

Name of an element in the specified library. Hyphens are also permitted. However, hyphens are not permitted as the last character.

VERSION = *HIGHEST-EXISTING / *STD / <composed-name 1..24>

Addition of the version to the element name. The default value is the latest version of the specified element.

TYPE = S / <alphanum-name 1..8>

Type of the element.

S Element type S (source program); default value.

D Element type D (text data)

M Element type M (macro)

TO = *PRIMARY

Resets SYSDTA to the primary assignment (see [section “System files” on page 1-73](#)).

TO = *SYSCMD

Combines SYSDTA and SYSCMD, i.e. the system reads both commands and data via SYSCMD.

DATA-ESCAPE-CHAR =

This specifies whether variables are to be replaced in data records.



This operand is allowed only in procedure mode. It is not available in interactive mode and guided dialog.

DATA-ESCAPE-CHAR = *COMPATIBLE

The replacement of variables and expressions in data records is to be performed in compatible fashion with the previous behavior:

- When the assignment SYSDTA not equal to SYSCMD is made, no replacement takes place in data records
- When SYSDTA=SYSCMD, replacement takes place in data records as defined in the SET-PROCEDURE-OPTIONS, MODIFY-PROCEDURE-OPTIONS or BEGIN-PROCEDURE command.

DATA-ESCAPE-CHAR = '&&' / '#' / '*' / '@' / '\$' / *STD

*This specification is possible only with the assignment SYSDTA not equal to SYSCMD. In the case of TO=*SYSCMD or TO=*PRIMARY, the specification is rejected.*

In data records, variables and expressions which begin with the specified character are to be replaced. Permissible characters are &, #, *, @ and \$. Specifying & is equivalent to specifying *STD (as on command level). If the character & is to be specified explicitly, it *must* be entered twice.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
2	0	SSM3200	Command executed with warning
2	0	SSM3034	The system file is already assigned to *PRIMARY
	1	SSM2036	Operand invalid
	32	SSM1013	System error during command execution
	64	SSM1025	Floppy disk device not available
	64	SSM1026	Floppy disk device not available or the floppy disk could not be mounted
	64	SSM2061	Error on accessing PLAM library element
	64	SSM2064	System file cannot be assigned via RFA
	64	SSM3030	SYSDTA is already assigned to floppy disk
	64	SSM3031	Semantic error
	64	SSM3055	Invalid record or file format
	64	SSM3056	OPEN error on input (DMS error is contained as in insert in the SYSOUT message)
	64	SSM3100	S variable does not exist
	64	SSM3101	S variable not of list type
	64	SSM3102	SDF-P subsystem not available
	64	SSM3104	DATA-ESCAPE-CHAR not equal to *COMPATIBLE not permitted when SYSDTA=SYSCMD
	64	SSM3105	Specification of DATA-ESCAPE-CHAR not valid in the current software configuration
	64	SSM3200	Error on accessing PLAM library or element
	65	SSM2074	SPOOL subsystem not loaded; a floppy disk device cannot be accessed

Notes

- Only one file may be assigned with one ASSIGN-SYSDTA command. Users wishing to assign more files must enter a corresponding number of ASSIGN-SYSDTA commands.
- At the procedure level, the system file SYSDTA is no longer assigned when the end of the file has been reached. Any further reading results in the message “SYSDTA NOT ASSIGNED”. When processing returns to a lower procedure level, SYSDTA is assigned to the device that was defined at that lower level.
- The variables replacement which is defined in the DATA-ESCAPE-CHAR operand for SYSDTA not equal to SYSCMD is restricted to S procedures.
The following applies for this setting:
 - When data in the context of a non-S procedure or ENTER file is read, the read operation is rejected with message SSM3106.
 - If an error occurs while variables are being replaced, the data record is rejected, and reading from the input source is terminated with EOF.
 - If the primary allocation is switched to while reading from the input source (file, library member or S variable), no variables are replaced (e.g. EDT switches to primary allocation after EOF occurs).

Examples

Example 1

```

/start-assembh _____ (1)
% BLS0523 ELEMENT 'ASSEMBH', VERSION '013', TYPE 'C' FROM LIBRARY ':10SH:$TSOS.
SYSPRG.ASSEMBH.013' IN PROCESS
% BLS0500 PROGRAM 'ASSEMBH', VERSION '01.3A02' OF '2012-04-04' LOADED
% BLS0552 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2012. ALL RIGHTS RESERVED
% ASS6010 V01.3A02 OF BS2000 ASSEMBH READY
%
//compile source=*sysdta,...
.
.
% ASS6012 END OF ASSTRAN
/assign-sysdta to=src.testprog _____ (2)
/start-assembh
% BLS0523 ELEMENT 'ASSEMBH', VERSION '013', TYPE 'C' FROM LIBRARY ':10SH:$TSOS.
SYSPRG.ASSEMBH.013' IN PROCESS
% BLS0500 PROGRAM 'ASSEMBH', VERSION '01.3A02' OF '2012-04-04' LOADED
% BLS0552 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2012. ALL RIGHTS RESERVED
% ASS6010 V01.3A02 OF BS2000 ASSEMBH READY
% ASS6011 ASSEMBLY TIME: 385 MSEC
% ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
% ASS6019 HIGHEST ERROR-WEIGHT: WARNING
% ASS6006 LISTING GENERATOR TIME: 121 MSEC
% ASS6012 END OF ASSTRAN
/assign-sysdta to=*primary _____ (3)

```

- (1) The system file SYSDTA has primary assignment. The assembler reads its input from the terminal.
- (2) SYSDTA is assigned to the file *SRC.TEST.PROG*. The assembler reads its input from this file (COMPILE statement and source program if SOURCE=*SYSDTA is declared in the COMPILE statement).
- (3) The primary assignment is returned to SYSDTA.

Example 2

```
/assign-sysdta to=*lib-elm(lib=biblio,elm=testprog)
/start-asmh
% BLS0523 ELEMENT 'ASSEMBH', VERSION '013', TYPE 'C' FROM LIBRARY ':10SH:$TSOS.
SYSPRG.ASSEMBH.013' IN PROCESS
% BLS0500 PROGRAM 'ASSEMBH', VERSION '01.3A02' OF '2012-04-04' LOADED
% BLS0552 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2012. ALL RIGHTS RESERVED
% ASS6010 V01.3A02 OF BS2000 ASSEMBH READY
% ASS6011 ASSEMBLY TIME: 429 MSEC
% ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
% ASS6019 HIGHEST ERROR-WEIGHT: WARNING
% ASS6006 LISTING GENERATOR TIME: 154 MSEC
% ASS6012 END OF ASSTRAN
/assign-sysdta to=*primary
```

The assembler reads its input from the element *TESTPROG* from the library file *BIBLIO*. After the compiler run has been completed, SYSDTA once more obtains the primary assignment.

Example 3

```
/BEG-PROC LOG=*ALL, PAR=*YES(PROC-PAR=(&INFILE1,&OUTFILE,&INFILE2,&SFID))
/ASS-SYSDTA TO=*SYSCMD
/START-SDF-I
OPEN INPUT-FILE=&INFILE1, OUTPUT-FILE=&OUTFILE
MERGE FILE=&INFILE2, REMOVE-ID=&SFID
END
/END-PROC
```

Because of the ASSIGN-SYSDTA command, when the non-S procedure executes SYSDTA, like SYSCMD, is contained in the file in which the procedure is stored. This then also enables the SDF-I utility to read the statements from this procedure file.

ASSIGN-SYSEVENT

Assign event stream to user task

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Job processing Operator terminal control
Domain:	CONSOLE-MANAGEMENT JOB
Privileges:	STD-PROCESSING OPERATING

Function

This command causes an event stream to be assigned to the user task from which it was issued. It has essentially three main functions:

1. Assigning an event stream to the user's own task. This registers the user task with the event stream service (ESS) and causes an event stream task to start event stream logging in an event stream file (ESF). The components of the event stream are all asynchronous events addressed to the user task and the synchronous events defined when the command is issued. A nonprivileged user is allowed to assign a user event stream (TO=*USER-LOG). The user task is connected to a private user event stream, and events are logged in a private user event stream file (UESF). A user with the OPERATING privilege is able to assign a system event stream (TO=*SYSTEM-LOG), which involves the logging of all asynchronous system messages distributed via routing codes. In such cases the user task is connected to the central system event stream, and events are logged in the central system event stream file (SESF).

A user task can only ever be assigned one event stream (user or system event stream). If a user task with an event stream already assigned to it is assigned a new event stream, the assignment is changed to match the new specifications in the command. The assignment which applied before is canceled (implied TO=*DUMMY).

2. Canceling the assignment of an event stream to the user's own task (ASSIGN-SYSEVENT TO=*DUMMY). The user task is deregistered from the ESS, and the events from the event stream are no longer logged in the ESF.
3. Modifying the assigned event stream. This primarily lets you redefine the set of synchronous messages being logged within an existing assignment (i.e. for your own active event stream).

Format

<p>ASSIGN-SYSEVENT</p> <p>TO = *USER-LOG (...) / *DUMMY / *CURRENT / *SYSTEM-LOG(...)</p> <p> *USER-LOG(...)</p> <p> LOG-ID = *OWN-TSN / <alphanum-name 1..4></p> <p> , OPEN-MODE = *CREATE / *OUTPUT / *EXTEND</p> <p> *SYSTEM-LOG(...)</p> <p> LOG-ID = *OWN-TSN / <alphanum-name 1..4></p> <p> , OPEN-MODE = *CREATE / *OUTPUT / *EXTEND</p> <p> ,ADD-SYNCH-EVENTS = *UNCHANGED-OR-STD / *NONE / *ALL / list-poss(3): *SYSOUT-MSG / *CMD / *STMT</p> <p> ,CLOSE-MODE = *UNCHANGED-OR-STD / *DELETE-EVENTS / *KEEP-EVENTS</p>

Operands

TO = *USER-LOG(...) / *DUMMY / *CURRENT / *SYSTEM-LOG(...)

Specifies the logical medium to which the assignment is to relate, thereby defining the function of the command (start, stop or modify an event stream).

TO = *USER-LOG(...)

Assigns a user event stream to the user task. The associated events are logged in a UESF.

LOG-ID = *OWN-TSN / <alphanum-name 1..4>

Defines the name that is assigned to the event stream. The default name is the TSN of your own task.

OPEN-MODE = *CREATE / *OUTPUT / *EXTEND

Defines the open mode for event stream assignment. You can either create a new event stream or continue with an existing one.

OPEN-MODE = *CREATE

Creates and starts a new event stream. The UESF is created. If an event stream of the same name and type already exists, the command is rejected.

OPEN-MODE = *OUTPUT

Reassigns and overwrites an existing, closed event stream. If the event stream does not yet exist, a new one is created. The UESF is overwritten, or a new one is created. The event stream record is updated accordingly.

OPEN-MODE = *EXTEND

Reassigns and extends an existing, closed event stream. If the event stream does not exist, the command is rejected.

TO = *DUMMY

Cancels the currently applicable assignments of the event stream. Logging of the associated events in the UESF or the central SESF is terminated. As you can only refer to your own active event stream, there is no need to indicate a LOG-ID.

TO = *CURRENT

Modifies the properties of the currently applicable event stream assignment. As you can only refer to your own active event stream, there is no need to indicate a LOG-ID.

TO = *SYSTEM-LOG(...)

Reserved for users with the OPERATING privilege. Assigns a system event stream to the user task. The associated events are logged in the central SESF.

LOG-ID = *OWN-TSN / <alphanum-name 1..4>

Defines the name that is assigned to the event stream. The default name is the TSN of your own task.

OPEN-MODE = *CREATE / *OUTPUT / *EXTEND

Defines the open mode for event stream assignment. You can either create a new event stream or continue with an existing one.

OPEN-MODE = *CREATE

Creates and starts a new event stream. If an event stream of the same name and type already exists, the command is rejected.

OPEN-MODE = *OUTPUT

Reassigns and overwrites an existing, closed event stream. If the event stream does not yet exist, a new one is created. The event stream record is updated accordingly.

OPEN-MODE = *EXTEND

Reassigns and extends an existing, closed event stream. If the event stream does not exist, the command is rejected.

ADD-SYNCH-EVENTS = *UNCHANGED-OR-STD / *NONE / *ALL / list-poss(3): *SYSOUT-MSG / *CMD / *STMT

Defines the logical set of event stream components that is to be logged in addition to the asynchronous messages addressed to the user task.

ADD-SYNCH-EVENTS = *UNCHANGED-OR-STD

Additional logging is determined by the type of assignment:

If you are assigning an event stream using TO=*USER-LOG or *SYSTEM-LOG, then ADD-SYNCH-EVENTS=*NONE applies. If you are modifying the assigned event stream using TO=*CURRENT, the current setting is retained unmodified. The operand is ignored for an assignment using TO=*DUMMY.

ADD-SYNCH-EVENTS = *NONE

Only asynchronous events addressed to the user task are to be logged.

ADD-SYNCH-EVENTS = *ALL

All message types are to be logged.

ADD-SYNCH-EVENTS = *SYSOUT-MSG

Messages sent to SYSOUT by command servers and user programs are to be logged in addition to asynchronous messages. Logging also extends to:

- all system messages and unformatted WROUT output, but not to FHS masks or other formatted screen output (such as EDT masks)
- all WRTRD output of the message system (MIP) which requires a response. A message requiring a response (query message) is always logged together with the response because the query and the response are logically related.

ADD-SYNCH-EVENTS = *CMD

Commands input from the user task which are executed using SDF are to be logged in addition to the asynchronous messages.

ADD-SYNCH-EVENTS = *STMT

Statements input from within a user program in the user task which are executed using SDF are to be logged in addition to the asynchronous messages.

CLOSE-MODE = *UNCHANGED-OR-STD / *DELETE-EVENTS / *KEEP-EVENTS

Governs whether the event stream is simply to be closed or is also to be deleted when the assignment is canceled. If the assignment is switched from one event stream to another, this operand only ever applies to the second event stream. The event stream that is closed when the assignment is switched keeps the setting it had beforehand.

CLOSE-MODE = *UNCHANGED-OR-STD

If you are assigning the event stream using LOG-ID, this operand value defaults to CLOSE-MODE=*DELETE-EVENTS. If you are canceling the assignment using TO=*DUMMY or modifying the assigned event stream using TO=*CURRENT, the setting defaults to *UNCHANGED, which means that the previous setting is retained.

CLOSE-MODE = *DELETE-EVENTS

Specifies that the event stream record and the logged events are to be deleted when the assignment is canceled. If a user task has been assigned a user event stream (TO=*USER-LOG), the associated UESF is deleted as well.

CLOSE-MODE = *KEEP-EVENTS

Specifies that the event stream record and the logged events are not to be deleted when the assignment is canceled. The SHOW-SYSEVENT-LOG command can still be used to display the logged events. If a user task has been assigned a user event stream (TO=*USER-LOG), the associated UESF is saved.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed successfully
1	0	NBR3103	No need to execute command, no event stream assigned
	1	CMD0202	Syntax error
	32	NBR3199	Internal error on command server
	32	NBR3111	Error generating user event stream task (UEST)
	32	NBR3112	System event stream task (SEST) not available
	64	CMD0216	User does not have the necessary privilege
	64	NBR3012	Initialization error in user event stream task (UEST)
	64	NBR3014	Termination error in user event stream task (UEST)
	64	NBR3102	Name of event stream not unique
	64	NBR3106	Event stream already assigned, OPEN-MODE=*CREATE operand rejected
	64	NBR3107	Event stream file (ESF) locked
	64	NBR3108	Event stream not available
	64	NBR3113	Event stream not assigned owing to /SHUTDOWN

ASSIGN-SYSLST

Assign SYSLST to output destination

Description status:	SYSFILE V19.0A
Functional area:	Job processing File processing
Domain:	JOB PROCEDURE PROGRAM
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

The ASSIGN-SYSLST command assigns a SYSLST system file to an output destination. SYSLST system files are: SYSLST, SYSLST01, SYSLST02,...SYSLST99. The primary assignment and attributes of SYSLST system files are described in [section "System files" on page 1-73](#).

Restrictions

Allocating a system file to a variable with TO=*VARIABLE(...) is only possible if the chargeable subsystem SDF-P is loaded.

Format

ASSIGN-SYSLST

```

TO = <filename 1..54> / *VARIABLE(...) / *LIBRARY-ELEMENT(...) / *PRIMARY / *DUMMY /
      *SYSLST-NUMBER(...)
  *VARIABLE(...)
    | VARIABLE-NAME = <composed-name 1..254>
  *LIBRARY-ELEMENT(...)
    | LIBRARY = <filename 1..54 without-gen-vers>
    | ELEMENT = <composed-name 1..64 with-under>(…)
    |               <composed-name 1..64 with-under>(…)
    |               | VERSION = *UPPER-LIMIT / <composed-name 1..24>
    | TYPE = P / *STD / <alphanum-name 1..8>
  *SYSLST-NUMBER(…)
    | SYSLST-NUMBER = <integer 1..99>
  ,OPEN-MODE = *OUTPUT / *EXTEND
  ,SYSLST-NUMBER = *STD / <integer 1..99>
  ,CODED-CHARACTER-SET = *TASK-TYPE-DEFAULT / *NONE / <name 1..8>

```

Operands

TO =

Output destination to which the SYSLST file is to be assigned.

TO = <filename 1..54>

Name for the new file or name of an already existing file to which the SYSLST file is to be assigned. A new file is created with the following attributes:

- SAM access method
- Initial allocation of the size defined by the system parameter SSMAPRI (default: 24 PAM pages).
- Secondary allocation of the size defined by the system parameter SSMASEC (default: 24 PAM pages).

An existing file must have the following attributes:

- SAM access method
- it must not reside on a multfile tape.

For SYSLST (not SYSLST01 through SYSLST99):

If no further storage space is available during SYSLST output to a disk file, in batch mode the system automatically requests a tape, copies the file to this tape and deletes the disk file. The SYSLST output is then continued in the tape file.

In interactive mode the system asks the user (message SSM2035) how to proceed further. The user has the following response options:

- T SYSLST output to a tape file (as in batch mode)
- M SYSLST output to MTC (procedure as for tape file)
- P Close the assigned file and change assignment to the primary assignment
- N Close the assigned file and change assignment to the dummy file *DUMMY (thereby losing further outputs!)

In the event of an error, the assignment is returned to the primary assignment.

Name of the file on tape or MTC: name of the disk file

TO = *VARIABLE(...)

Name of a complex S variable of the "list" type (i.e. a series of elements of the same variable type which can only be accessed sequentially). Complex S variables can only be used if the chargeable subsystem SDF-P is loaded (see the "SDF-P" manual [34]). The command will be rejected if the specified list has not been declared or if the list elements are not of variable type STRING (or ANY).

VARIABLE-NAME = <composed-name 1..254>

Name of the S variable.

TO = *LIBRARY-ELEMENT(...)

Output to a PLAM library element.

LIBRARY = <filename 1..54 without-gen-vers>

Name of a PLAM library.

ELEMENT = <composed-name 1..64 with-under>(...)

Name of an element in the specified library. Hyphens are also permitted. However, hyphens are not permitted as the last character.

VERSION = *UPPER-LIMIT / <composed-name 1..24>

Addition of the version to the element name. The default value is the highest possible version of the specified element.

TYPE = P / *STD / <alphanum-name 1..8>

Type of the element. The *STD operand value corresponds to element type P .

P	Element type P (print-edited data); default value
S	Element type S (source program)
D	Element type D (text data)
M	Element type M (macro)

For SYSLST (not SYSLST01 through SYSLST99):

If no further storage space is available during SYSLST output to a library element, in batch mode the system automatically requests a tape, copies the element to this tape and deletes it from the PLAM library. The SYSLST output is then continued in the tape file. In interactive mode the system asks the user (message SSM2035) how to proceed further.

The user has the following response options:

T	SYSLST output to a tape file (as in batch mode)
M	SYSLST output to MTC (procedure as for tape file)
P	Close the assigned library element and change assignment to the primary assignment
N	Close the assigned library element and change assignment to the dummy file *DUMMY (thereby losing further outputs!)

In the event of an error, the assignment is returned to the primary assignment.

Name of the file on tape or MTC:

S.LST.PLAM-T0-TAPE.<tsn>.<date>.<time>

<tsn> = TSN of the task

<date> = Creation date in the form yyyy-mm-dd, where yyyy=year, mm=month and dd=day

<time> = Creation time in the form hhmmss, where hh=hours, mm=minutes and ss=seconds

TO = *PRIMARY

Resets the SYSLST file to its primary assignment (see [section "System files" on page 1-73](#)).

TO = *DUMMY

Assigns the SYSLST file to a dummy file, which means that output to a volume is suppressed. For further information on using dummy files see the FILE-NAME=*DUMMY operand in the ADD-FILE-LINK command description.

TO = *SYSLST-NUMBER(...)

Designates as output destination a SYSLST file from the series SYSLST01 through SYSLST99, to which a SYSLST file from the series SYSLST01 through SYSLST99 is to be assigned. This operand value is not permitted when SYSLST-NUMBER = STD (the default) applies, i.e. the system file SYSLST must not be assigned to a system file in the series SYSLST01 to SYSLST99.

SYSLST-NUMBER = <integer 1..99>

Number of the SYSLST file. It must not be the same as that specified in the (subsequent) operand SYSLST-NUMBER.

OPEN-MODE = *OUTPUT / *EXTEND

Specifies whether an already cataloged SAM file is to be replaced or extended.

If a complex S variable is assigned:

If OPEN-MODE=*OUTPUT is specified, output resumes with the first list element. Previous list items can no longer be accessed. If OPEN-MODE=*EXTEND is specified, the list of variables is extended (dynamic extension must have been permitted).

SYSLST-NUMBER =

Designates the SYSLST file to which the specified output destination is to be assigned.

SYSLST-NUMBER = *STD

The system file SYSLST is assumed. This operand value must not be specified together with TO = *SYSLST-NUMBER

SYSLST-NUMBER = <integer 1..99>

The system file from the series SYSLST01 through SYSLST99 whose number is specified here is used. If the operand TO = *SYSLST-NUMBER is specified, the number given here must not correspond to the number specified there.

CODED-CHARACTER-SET = *TASK-TYPE-DEFAULT / *NONE / <name 1..8>

Determines the code of the SYSLST file. This specifies how the characters in a national character set are to be stored in binary form. The specified character set has an effect on the representation of characters on the screen, the collating sequence, etc. (see the "XHCS" manual [51]).

CODED-CHARACTER-SET = *TASK-TYPE-DEFAULT

The code defined for the SYSLST file depends on the task mode:

- No code table is used within a batch task (corresponds to CODED-CHARACTER-SET=*NONE).
- The code table currently set for the task is used within a dialog task (see also the function description of the MODIFY-TERMINAL-OPTIONS command).

CODED-CHARACTER-SET = *NONE

No code is defined for the SYSLST file.

CODED-CHARACTER-SET = <name 1..8>

The specified code is defined for the SYSLST file.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
1	0	SSM2039	Command executed but error occurred on closing the previously assigned SYSLST file (DMS error is contained as an insert in the SYSOUT message)
2	0	SSM3200	Command executed with warning
2	0	SSM3034	The system file is already assigned to *PRIMARY
	1	SSM2036	Operand invalid
	1	SSM2038	The specified code table (CCS) is not permitted
	32	SSM1013	System error during command execution
	64	SSM0001	Loop during SYSLSTnn assignment
	64	SSM2012	OPEN error on output (DMS error is contained as an insert in the SYSOUT message)
	64	SSM2048	XHCS does not support the specified code table (CCS)
	64	SSM2061	Error on accessing PLAM library element
	64	SSM2064	System file cannot be assigned via RFA
	64	SSM3031	Semantic error
	64	SSM3055	Invalid record or file format
	64	SSM3100	S variable does not exist
	64	SSM3101	S variable not of list type
	64	SSM3102	SDF-P subsystem not available
	64	SSM3200	Error on accessing PLAM library or element

Note

No loops may arise in the assignment of SYSLST files to one another. For example the following combination is rejected with an error message:

```

/ASS-SYSLST... SYSLSTxx  ──────────> SYSLSTyy
/ASS-SYSLST... SYSLSTyy  ──────────> SYSLSTxx
    
```

Examples

Example 1

```

/ass-syslst to=*dummy _____ (1)
/show-sys-file system-file=*syslst
%PROCEDURE LEVEL NUMBER 0
%SYSLST : *DUMMY
/start-exe from-file=mj.cob.dat
.
.
.
/ass-syslst to=mj.cob.dat,open-mode=*ext _____ (2)
/show-sys-file system-file=*syslst
%PROCEDURE LEVEL NUMBER 0
%SYSLST : :P:$USER.MJ.COB.DAT
/start-exe from-file=cob.prog
.
.
.
/ass-syslst to=*primary _____ (3)

```

- (1) A user program prints data on the printer. The user does not require any listings for trial operation, assigning SYSLST to a dummy file instead.
- (2) In productive use, the data is not to be output on the printer either, but is instead to be appended to the existing file MJ.COB.DAT.
- (3) SYSLST again receives its primary assignment.

Example 2 (Assigning SYSLST-NUMBER)

```

/ass-syslst to=lst.syssdf.globals,syslst-number=1 _____ (1)
/ass-syslst to=lst.syssdf.implementation,syslst-number=2 _____ (2)
/ass-syslst to=lst.syssdf.cmd-list,syslst-number=3 _____ (3)
/show-sys-file _____ (4)
%PROCEDURE LEVEL NUMBER 0
%SYSCMD : (PRIMARY)
%SYSDTA : (PRIMARY)
%SYSIPT : NOT ASSIGNED
%SYSOUT : (PRIMARY)
%SYSLST : (PRIMARY)
%SYSLST01: :IOSN:$USERXY01.LST.SYSSDF.GLOBALS
%SYSLST02: :IOSN:$USERXY01.LST.SYSSDF.IMPLEMENTATION
%SYSLST03: :IOSN:$USERXY01.LST.SYSSDF.CMD-LIST
%SYSOPT : (PRIMARY)
%TASKLIB : (PRIMARY)
/start-sdf-a _____ (5)
% BLS0517 MODULE 'SDAMAIN' LOADED
% SDA0001 'SDF-A' VERSION '04.1G10' STARTED
%/open SYSSDF.NDCSFSYS,type=*system
%/show object=*global-info,output=*syslst(syslst-number=1)
%/show object=*cmd(name=*all),impl-info=*yes,output=*syslst(2)
%/show object=*cmd(name=*all),attach-info=*no,output=*syslst(3)
%/end
/ass-syslst to=*primary,syslst-number=1 _____ (6)
/ass-syslst to=*primary,syslst-number=2
/ass-syslst to=*primary,syslst-number=3

```



```

/show-sys-file
%PROCEDURE LEVEL NUMBER 0
%SYSCMD : (PRIMARY)
%SYSDTA : (PRIMARY)
%SYSIPT : NOT ASSIGNED
%SYSOUT : (PRIMARY)
%SYSLST : (PRIMARY)
%SYSOPT : (PRIMARY)
%TASKLIB : (PRIMARY)

```

- (1) System file SYSLST01 is assigned to *LST.SYSSDF.GLOBALS*.
- (2) System file SYSLST02 is assigned to *LST.SYSSDF.IMPLEMENTATION*.
- (3) System file SYSLST03 is assigned to *LST.SYSSDF.CMD-LIST*.
- (4) The SHOW-SYSTEM-FILE-ASSIGNMENT command shows the assignments of the system files.
- (5) Call to the SDF-A utility routine (syntax file processing).
 Various items of information are to be output to files from a syntax file (global information, implementation statements and a command list). The SDF-A SHOW statement only supports output to SYSOUT, SYSLST or SYSLST-NUMBER. To avoid having to keep changing the SYSLST assignment while the program is running, a number of SYSLST-NUMBER files were assigned to files before the program was called. Now a specific SYSLST-NUMBER file can be selected in the SHOW statements and the output sent to the file assigned to it.
- (6) Once the program has executed, the assignment are canceled (see the output of the SHOW-SYSTEM-FILE-ASSIGNMENT command).

ASSIGN-SYSOUT

Assign SYSOUT to output destination

Description status:	SYSFILE V19.0A
Functional area:	Job processing File processing
Domain:	JOB PROCEDURE PROGRAM
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

The ASSIGN-SYSOUT command assigns the system output file SYSOUT to an output destination. In interactive mode SYSOUT is assigned primarily to the terminal. If it is reassigned to a different output destination, *all* outputs are redirected. In addition, output may be displayed on the terminal (TERMINAL-DISPLAY operand set to *YES). If this option is not taken (TERMINAL-DISPLAY=*NO), the user's outputs are not restored to the terminal until SYSOUT is given the assignment *PRIMARY.

The primary assignment and attributes of SYSOUT are described in [section "System files" on page 1-73](#).

Restrictions

Allocating a system file to a variable with TO=*VARIABLE(...) is only possible if the chargeable subsystem SDF-P is loaded.

Format

ASSIGN-SYSOUT

```

TO = <filename 1..54> / *VARIABLE(...) / *LIBRARY-ELEMENT(...) / *PRIMARY / *DUMMY
  *VARIABLE(...)
    |   VARIABLE-NAME = <composed-name 1..254>
  *LIBRARY-ELEMENT(...)
    |   LIBRARY = <filename 1..54 without-gen-vers>
    |   ELEMENT = <composed-name 1..64 with-under>(…)
    |               <composed-name 1..64 with-under>(…)
    |               |   VERSION = *UPPER-LIMIT / <composed-name 1..24>
    |               |   TYPE = P / *STD / <alphanum-name 1..8>
  ,OPEN-MODE = *OUTPUT / *EXTEND
  ,TERMINAL-DISPLAY = *NO / *YES
  ,CODED-CHARACTER-SET = *TASK-TYPE-DEFAULT / *NONE / <name 1..8>

```

Operands

TO =

Output destination to which SYSOUT is to be assigned.

TO = <filename 1..54>

Name for the new file or name of an existing file to which SYSOUT is to be assigned. A new file is created with the following attributes:

- SAM access method
- Initial allocation of the size defined by the system parameter SSMAPRI (default: 24 PAM pages).
- Secondary allocation of the size defined by the system parameter SSMASEC (default: 24 PAM pages).

If an existing file is to be used, it must be a SAM file.

If no further storage space is available during SYSOUT output to a disk file, in batch mode the system automatically requests a tape, copies the file to this tape and deletes the disk file. The SYSOUT output is then continued in the tape file.

In interactive mode the system asks the user (message SSM2035) how to proceed further. The user has the following response options:

- T SYSOUT output to a tape file (as in batch mode)
- M SYSOUT output to MTC (procedure as for tape file)
- P Close the assigned file and change assignment to the primary assignment
- N Close the assigned file and change assignment to the dummy file *DUMMY (thereby losing further outputs!)

In the event of an error, the assignment is returned to the primary assignment. Name of the file on tape or MTC: name of the disk file

TO = *VARIABLE(...)

Name of a complex S variable of the "list" type (i.e. a series of elements of the same variable type which can only be accessed sequentially). Complex S variables can only be used if the chargeable subsystem SDF-P is loaded (see the "SDF-P" manual [34]). The command will be rejected if the specified list has not been declared or if the list elements are not of variable type STRING (or ANY).

VARIABLE-NAME = <composed-name 1..254>

Name of the S variable.

TO = *LIBRARY-ELEMENT(...)

Output to a PLAM library element.

LIBRARY = <filename 1..54 without-gen-vers>

Name of a PLAM library.

ELEMENT = <composed-name 1..64 with-under>(...)

Name of an element in the specified library. Hyphens are also permitted. However, hyphens are not permitted as the last character.

VERSION = *UPPER-LIMIT / <composed-name 1..24>

Addition of the version to the element name. The default value is the highest possible version of the specified element.

TYPE = P / *STD / <alphanum-name 1..8>

Type of the element. The *STD operand value corresponds to element type P .

- P Element type P (print-edited data); default value
- S Element type S (source program)
- D Element type D (text data)
- M Element type M (macro)

If no further storage space is available during SYSOUT output to a library element, in batch mode the system automatically requests a tape, copies the element to this tape and deletes it from the PLAM library. The SYSOUT output is then continued in the tape file. In interactive mode the system asks the user (message SSM2035) how to proceed further. The user has the following response options:

- T SYSOUT output to a tape file (as in batch mode)
M SYSOUT output to MTC (procedure as for tape file)
P Close the assigned library element and change assignment to the primary assignment
N Close the assigned library element and change assignment to the dummy file *DUMMY (thereby losing further outputs!)

In the event of an error, the assignment is returned to the primary assignment.

Name of the file on tape or MTC:

S.OUT.PLAM-T0-TAPE.<tsn>.<date>.<time>

<tsn> = TSN of the task

<date> = Creation date in the form yyyy-mm-dd, where yyyy=year, mm=month and dd=day

<time> = Creation time in the form hhmmss, where hh=hours, mm=minutes and ss=seconds

TO = *PRIMARY

Resets SYSOUT to its primary assignment (see [section "System files" on page 1-73](#)).

TO = *DUMMY

Assigns SYSOUT to a dummy file, which means that output to a volume is suppressed. For further information on using dummy files see the FILE-NAME=*DUMMY operand in the ADD-FILE-LINK command description.

OPEN-MODE = *OUTPUT / *EXTEND

Specifies whether an already cataloged SAM file is to be replaced or extended. If a complex S variable is assigned: If OPEN-MODE=*OUTPUT is specified, output resumes with the first list element. Previous list items can no longer be accessed. If OPEN-MODE=*EXTEND is specified, the list of variables is extended (dynamic extension must have been permitted).

TERMINAL-DISPLAY = *NO / *YES

Specifies whether output to SYSOUT should also be displayed on the terminal. This operand is allowed only in interactive mode. If SYSOUT has its primary assignment, the operand is ignored.

CODED-CHARACTER-SET = *TASK-TYPE-DEFAULT / *NONE / <name 1..8>

Determines the code of the SYSOUT file. This specifies how the characters in a national character set are to be stored in binary form. The specified character set has an effect on the representation of characters on the screen, the collating sequence, etc. (see the "XHCS" manual [51]).

CODED-CHARACTER-SET = *TASK-TYPE-DEFAULT

The code defined for the SYSOUT file depends on the task mode:

- No code table is used within a batch task (corresponds to CODED-CHARACTER-SET=*NONE).
- The code table currently set for the task is used within a dialog task (see also the function description of the MODIFY-TERMINAL-OPTIONS command).

CODED-CHARACTER-SET = *NONE

No code is defined for the SYSOUT file.

CODED-CHARACTER-SET = <name 1..8>

The specified code is defined for the SYSOUT file.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
1	0	SSM2039	Command executed but error occurred on closing the previously assigned SYSLST file (DMS error is contained as an insert in the SYSOUT message)
2	0	SSM3200	Command executed with warning
2	0	SSM3034	The system file is already assigned to *PRIMARY
	1	SSM2036	Operand invalid
	1	SSM2038	The specified code table (CCS) is not permitted
	32	SSM1013	System error during command execution
	64	SSM2012	OPEN error on output (DMS error is contained as an insert in the SYSOUT message)
	64	SSM2048	XHCS does not support the specified code table (CCS)
	64	SSM2061	Error on accessing PLAM library element
	64	SSM2064	System file cannot be assigned via RFA
	64	SSM3031	Semantic error
	64	SSM3055	Invalid record or file format
	64	SSM3100	S variable does not exist
	64	SSM3101	S variable not of list type
	64	SSM3102	SDF-P subsystem not available
	64	SSM3200	Error on accessing PLAM library or element

Example

The ENTER file E.Entering contains the following records:

```
/SET-LOGON-PAR
/ASS-SYSOUT TO=L.PROT
/SHOW-SYS-FILE SYSTEM-FILE=*SYSOUT
/ASS-SYSOUT TO=*PRIMARY
/SHOW-SYS-STAT
/EXIT-JOB
```

Calling the ENTER file:

```
/enter-job from-file=e.enter,cpu-limit=19
% JMS0066 JOB '(NONE)' ACCEPTED ON 04-12-16 AT 14:55, TSN = 6VCN
```

During ENTER file execution the following records of the SYSOUT log are output

– on the printer:

```
/SET-LOGON-PAR
%....
/ASS-SYSOUT TO=L.PROT
/SHOW-SYS-STAT
% SCP1095 DPRINTSV WARNING : SOME DPRINT PRINT-JOBS CANNOT BE DISPLAYED
% T1 T2 T3 T4PR T4FD T4TP T5AC T5KP T6 T7 T8
% 0 1 1 6 0 0 0 0 0 0 0
/EXIT-JOB
%....
```

– to the file L.PROT:

```
/SHOW-SYS-FILE SYSTEM-FILE=SYSOUT
PROCEDURE LEVEL NUMBER 0
SYSOUT : :20SG:$USER1.L.PROT
/ASS-SYSOUT TO=F=*PRIMARY
```

ASTOP

Suspend command file

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Editing command files
Domain:	not allocated
Privileges:	OPERATING
Routing code:	@

Function

The ASTOP command suspends the processing of a command file. Processing of the command file is always resumed once the wait time set using the NBRUNWT system parameter has elapsed, even if an insufficient number of AGOGO commands has been specified.

For a description of how the command operates see “Command files for the operator” in the “Introduction to System Administration” [14].

The ASTOP command is allowed in RUN command files only. It is available in ISP format only; in other words, there are no SDF functions (such as syntax analysis or help).

Format

ASTOP
<u>1</u> / <integer 1..255>

Operands

1 / <integer 1..255>

Number of AGOGO commands that must be received before command file processing is continued. Default value: 1 AGOGO command must be received.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors

ATTACH-DEVICE

Attach hardware units

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Device management
Domain:	DEVICE
Privileges:	OPERATING
Routing code:	G

Function

This command enables the operator to make available one or more hardware units, i.e. to allow the operating system to use these units for I/O operations. The hardware units are attached via SVP.

Effect of the ATTACH-DEVICE command

1. If the specified hardware units were in the “detached explicitly” state, they are placed in the “attached” state and can be used again.
2. If the specified hardware units were in the “detached implicitly” state, an ATTACH is rejected. The “detached explicitly” state, if applicable, is canceled.
3. If the outward connections associated with the unit were in the “removed implicitly” state, they are placed in the “included” state. The connections can be used again.
4. All outward hardware units in the “detached implicitly” state which have at least one connection to the relevant unit in the “included” state are placed in the “attached” state. These units can be used again.
5. If there are hardware attach actions (SVP actions) for the units to be attached, these actions are initiated. The hardware units addressed explicitly in the command are not placed in the “attached” state unless the hardware reactions resulting from SVP processing receive a positive acknowledgment, otherwise ATTACH is rejected.
6. In the case of disks and tape devices for which it is possible to set up path groups, such path groups are set up for all the available paths.
7. When a PAV (**P**arallel **A**ccess **V**olumes device is attached (see the “Introduction to System Administration” [14]), the following applies:

In the case of a base device, the associated alias devices are searched for and implicitly attached. Alias devices can only be explicitly attached using the ATTACH-DEVICE command when the associated base device is already attached.

Format

ATTACH-DEVICE

```

UNIT = *CPU(...) / *EXTRA-CPU(...) / *CHANNEL(...) / *CONTROLLER(...) /
        *CHANNEL-RANGE(...) / *DEVICE-RANGE(...) / *PUBSET-DEVICES(...) /
        list-poss(255): <alphanum-name 2..2> / <x-text 4..4>

*CPU(...)
  |   CPU-IDENTIFIER = list-poss(16): <x-text 2..2>

*EXTRA-CPU(...)
  |   CPU-IDENTIFIER = *ALL / *ANY / <x-text 2..2>

*CHANNEL(...)
  |   CHANNEL-PATH-ID = list-poss(16): <x-text 2..2>
  |   ,SCOPE = *OWN-SYSTEM-ONLY / *VM2000-GLOBAL

*CONTROLLER(...)
  |   CONTROLLER-UNIT = list-poss(16): <alphanum-name 2..2> / <x-text 4..4>
  |   ,SCOPE = *OWN-SYSTEM-ONLY / *VM2000-GLOBAL

*CHANNEL-RANGE(...)
  |   FROM = <x-text 2..2>
  |   ,TO = <x-text 2..2>
  |   ,SCOPE = *OWN-SYSTEM-ONLY / *VM2000-GLOBAL

*DEVICE-RANGE(...)
  |   FROM = <alphanum-name 2..2> / <x-text 4..4>
  |   ,TO = <alphanum-name 2..2> / <x-text 4..4>

*PUBSET-DEVICES(...)
  |   PUBSET = list-poss(255): <cat-id 1..4> / *BY-PUBRES-DEVICE(...)
  |   *BY-PUBRES-DEVICE(...)
  |   |   UNIT = list-poss(255): <alphanum-name 2..2> / <x-text 4..4>

```

Operands

UNIT =

Specifies the hardware units to be attached to the system.

UNIT = *CPU(...)

Specifies the CPU to be attached.

CPU-IDENTIFIER = list-poss(16): <x-text 2..2>

Specifies the CPU identifier.

UNIT = *EXTRA-CPU(...)

Specifies the extra CPU that is to be attached.

CPU-IDENTIFIER = *ALL / *ANY / <x-text 2..2>

Specifies the identifier of the extra CPU. *ANY attaches one available extra CPU. *ALL attaches all available extra CPUs.

UNIT = *CHANNEL(...)

Specifies the channel to be attached.

CHANNEL-PATH-ID = list-poss(16): <x-text 2..2>

Specifies the channel path ID of the channel. In the case of channels, the device name (MN) is the same as the channel path ID in hexadecimal representation.

SCOPE =

Specifies how the command is to be executed under VM2000.

SCOPE = *OWN-SYSTEM-ONLY

The command is only executed in the local system.

SCOPE = *VM2000-GLOBAL

If entered on the Monitor System (SUs x/390 and S servers), the command is executed on all guest systems. If entered on another guest system or on the Monitor System (SUs x86 and S servers), the command is rejected with the message NKR0178.

UNIT = *CONTROLLER(...)

Specifies the controller to be attached.

CONTROLLER-UNIT = list-poss(16): <alphanum-name 2..2> / <x-text 4..4>

Specifies the mnemonic device code (MN) of the controller to be attached.

SCOPE =

Specifies how the command is to be executed under VM2000.

SCOPE = *OWN-SYSTEM-ONLY

The command is only executed in the local system.

SCOPE = *VM2000-GLOBAL

If entered on the Monitor System (SUs x/390 and S servers), the command is executed on all guest systems. If entered on another guest system or on the Monitor System (SUs x86 and S servers), the command is rejected with the message NKR0178.

UNIT = *CHANNEL-RANGE(...)

Specifies a set of channels to be attached.

FROM = <x-text 2..2>

Specifies the channel path ID of the first channel in the set of channels to be attached.

TO = <x-text 2..2>

Specifies the channel path ID of the last channel in the set of channels to be attached. The following rules apply: $chn_1-id < chn_2-id$, and $chn_2-id - chn_1-id < 64$, i.e. a maximum of 64 channels can be attached at a time.

SCOPE =

Specifies how the command is to be executed under VM2000.

SCOPE = *OWN-SYSTEM-ONLY

The command is only executed in the local system.

SCOPE = *VM2000-GLOBAL

If entered on the Monitor System (SUs x/390 and S servers), the command is executed on all guest systems. If entered on another guest system or on the Monitor System (SUs x86 and S servers), the command is rejected with the message NKR0178.

In the case of channels, the device name (MN) is the same as the channel path ID in hexadecimal representation.

UNIT = *DEVICE-RANGE(...)

Defines the maximum of 256 devices, that can be attached. The mnemonic codes defined do not have to be contiguous. If no device is generated for one of the codes, processing continues with the next device code.

FROM = <alphanum-name 2..2> / <x-text 4..4>

Defines the mnemonic device code (MN) of the first device in the set of devices to be attached.

TO = <alphanum-name 2..2> / <x-text 4..4>

Defines the mnemonic device code (MN) of the last device in the set of devices to be attached.

UNIT = *PUBSET-DEVICES(...)

Specifies that a pubset's disks are to be attached. Since the mnemonic device names (MNs) of the associated disks are administered in the SVL of the system disk, the latter must have been entered in the MRSCAT. The entry is made whenever the pubset is imported or exported in the case of modifications to the composition of the pubset (see MODIFY-PUBSET-PROCESSING). Alternatively you can enter the system disk manually with the ADD-MASTER-CATALOG-ENTRY and MODIFY-MASTER-CATALOG-ENTRY commands.

In all cases, as many disks as possible are attached. If pubset disks are mirrored then the following should be noted:

- If mirroring is performed with DRV, both disks are always attached.
- In the case of mirroring in external disk storage systems (see the “SHC-OSD” User Guide [37]) only the standard disks are attached (source or normal unit). If the mirror disks are to be attached then the mirror disk (target or additional mirror unit) of the system disk (pubres) must be specified in the PUBSET operand.

PUBSET = list-poss(255): <cat-id 1..4> / *BY-PUBRES-DEVICE(...)

Identifies the pubset whose disks are to be attached. The pubset can be specified via its catalog ID or the device code of its system disk.

PUBSET = <cat-id 1..4>

Specifies the pubset's catalog ID. There must be a corresponding entry in the MRS catalog. If there is no entry then the disks can only be attached by specifying the system disk (see PUBSET=*BY-PUBRES-DEVICE).

PUBSET = *BY-PUBRES-DEVICE(...)

Specifies the pubset's system disk (pubres).

UNIT = list-poss(255): <alphanum-name 2..2> / <x-text 4..4>

Mnemonic device code (MN) of the pubres.

UNIT = list-poss(255): <alphanum-name 2..2> / <x-text 4..4>

Specifies the mnemonic device code (MN) if a device is to be attached.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
1	64	ETMRK..	Command execution faulty
2	64	ETMRK..	Command partially processed without error
4	64	NKR0...	Hardware unit was already attached
12	64	NKR0...	Internal check negative
16	64	NKR0...	Caller error
20	64	NKR0...	Software error

If there is an error in command termination, the maincode contains the message code of the message output during command processing. The command return codes with the ETMRK.. maincodes occur only if the hardware unit to be reconfigured is a CPU.

Examples

Attach the channels with the channel path IDs 10 through 1F

```
/ATTACH UNIT=*CHANNEL-RANGE(FROM=10,TO=1F)           or  
/ATT *CH-R(10,1F)
```

Attach devices D1, D2 and C1

```
/ATTACH-DEVICE UNIT=(C1,D1,D2)                       or  
/ATT (C1,D1,D2)
```

Attach CPU 1

```
/ATTACH-DEVICE UNIT=*CPU(CPU-IDENTIFIER=01)         or  
/ATT *CPU(01)
```

Attaching the pubset's disks with the catalog ID 2OSG

```
/ATTACH-DEVICE UNIT=*PUBSET-DEVICES(PUBSET=2OSG)   or  
/ATT *PUB-DEV(2OSG)
```

ATTACH-GS-UNIT

Make GS unit operational

Description status:	GSMAN V19.0A
Functional area:	Caching media control Global storage administration
Domain:	DEVICE
Privileges:	TSOS

Function

The basic configuration of global storage (GS) is defined on the SVP. It includes the number of GS units available and how they are attached. All changes to the basic configuration must be made on the SVP, and they do not take effect until the system is restarted. S servers offer additional functions to connect and disconnect GS units which can be used to handle hardware faults and for maintenance purposes. The operating system makes these functions available to systems support personnel through the ATTACH- and DETACH-GS-UNIT reconfiguration commands. In the event of serious hardware faults, a GS unit is implicitly detached by the operating system. The GS unit should not be reattached until maintenance has been completed.

The applicability of the reconfiguration commands depends on the GS configuration (one or two GS units), on how they are currently being used and on the mode of GS operation in a HIPLEX MSCF network (see also the section headed “The global storage (GS) medium” in the “Introduction to System Administration (BS2000/OSD V9.0)” [14]):

- In a network with shared GS, the nodes use GS as a shared resource (global GS operation).
- In a cluster without shared GS, the nodes can use their GS locally (local GS mode).

In global GS operation within an XCS network, the command is effective for all systems in the network.

Note the following with regard to local GS operation:

- Detached GS units (DETACHED state, see SHOW-GS-STATUS command) can as a rule be reattached online using the ATTACH-GS-UNIT command.
 - Exception* If a GS unit is in the ATTACHED state and a dual partition is in use with the attribute ATTACH-DUAL=FORBIDDEN, the command is rejected for the second GS unit; use of all dual partitions with ATTACH-DUAL=FORBIDDEN must be terminated first.
- If a GS unit containing data which is no longer up to date is the first to be attached online, the system issues a request for verification (EGC2110). The last GS unit to be detached will be reattached without query.

The following applies in all cases:

- When the second GS unit is attached, the metadata of the first GS unit (in the ATTACHED state) and all dual partitions containing valid data are copied to the second GS unit. This equalization process may take a number of minutes. During equalization, the dual partition is locked against writing. Concurrent reading is allowed.
- Before a system restart, detached GS units can be attached with the aid of the SVP.
- In parallel HIPLEX, the command is not accepted if there is as yet no GS unit with the status ATTACHED. A precondition is that all the systems in the XCS network are running BS2000/OSD V5.0. If any system in the XCS network is running BS2000/OSD-BC < V5.0, then the command is rejected with the message EGC2010.
- The command starts the specified GS unit in the local GS complex. If there are as yet no connections in the GS complex, then the GS unit is connected to all the GS servers in the GS complex. If the other GS unit is already connected to GS servers then the specified GS unit is connected to the servers that are connected to the other GS unit.

The SHOW-GS-STATUS command allows systems support personnel to check which GS units are in operation in the local system.

Format

ATTACH-GS-UNIT
GS-UNIT = <integer 1..2>

Operands

GS-UNIT = <integer 1..2>

Specifies the number of the GS unit which is to be reattached.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
1	0	EGC2001	GS unit is already assigned
	1	CMD0202	Syntax error
	32	CMD0221	Internal error
	64	CMD0216	No authorization to call command
	64	EGC0005	Command aborted by the user
	64	EGC0112	No GS available
	64	EGC2004	GS unit does not exist
	64	EGC2010	First GS unit cannot be attached
	64	EGC2011	ATTACH-DUAL=FORBIDDEN
	64	EGC2012	Partition in use
	64	EGC2013	Communication error
	64	EGC2014	SVP error
	64	EGC2015	Equalization error
	64	EGC2016	system error
	128	EGC0010	GSMAN subsystem is not ready
	128	EGC0110	Command temporarily not executable

BEGIN-PARAMETER-DECLARATION

Initiate parameter declarations

Description status:	SDF-P-BASYS V2.5E
Functional area:	Procedures
Domain:	PROCEDURE
Privileges:	STD-PROCESSING OPERATING HARDWARE-MAINTENANCE SECURITY-ADMINISTRATION SAT-FILE-MANAGEMENT SAT-FILE-EVALUATION

Function

The BEGIN-PARAMETER-DECLARATION command is an SDF-P control flow command. It is part of the header of an S procedure, and it identifies the beginning of the declaration section in this header. The procedure parameters are declared in the declaration section. There may be a SET-PROCEDURE-OPTIONS command before the declaration section. The END-PARAMETER-DECLARATION command terminates both the declaration section and the procedure header. Within the declaration section, only DECLARE-PARAMETER commands are permitted. If only one DECLARE-PARAMETER command is used, the BEGIN-PARAMETER-DECLARATION and END-PARAMETER-DECLARATION commands can be omitted. The declaration section is redundant if no procedure parameters are declared. The BEGIN-PARAMETER-DECLARATION command is also required when one or more OPERAND-VARIABLE-CONTAINER commands need to be inserted in the procedure header so as to initialize procedure parameters with permanent variables. Procedure parameters are stored as S variables and can therefore be queried and modified using SDF-P facilities.

Format

BEGIN-PARAMETER-DECLARATION

Return codes

The BEGIN-PARAMETER-DECLARATION command can be used within the procedure header of an S procedure. SDF-P detects errors in the procedure head during pre-analysis and terminates the procedure call. The command return codes can only occur if the command is used outside the procedure header.

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
	1	CMD0202	Syntax error
	1	SDP0118	Command in false context
	3	CMD2203	Incorrect syntax file
	32	CMD0221	System error (internal error)
	130	SDP0099	No further address space available

BEGIN-PROCEDURE

Begin procedure file and define procedure parameters

Description status:	SYSFILE V19.0A
Functional area:	Procedures
Domain:	PROCEDURE
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

This command can only be used for non-S procedures.

The BEGIN-PROCEDURE command is required as the first record in a non-S procedure. It defines symbolic parameters that are used in the subsequent commands in the procedure file. Symbolic parameters are variables which are given actual values while the procedure is executing. These values may be defined as follows:

- when the procedure is called: in the CALL-PROCEDURE command,
- when the procedure is created: in the BEGIN-PROCEDURE command (so-called predefinition: these values are used only if no values are specified in the CALL-PROCEDURE command),
- while the procedure is executing (prompting): The user is requested to enter a value during the dialog if no values were defined for the corresponding symbolic operands either in the CALL-PROCEDURE command or in the BEGIN-PROCEDURE command.

The BEGIN-PROCEDURE command may be used only in procedure files. In interactive mode it is rejected, and is therefore not offered as a menu option.

Format

```

BEGIN-PROCEDURE

LOGGING = *NO / *ALL / *DATA / *CMD
,PARAMETERS = *NO / *YES(...)
  *YES(...)
    |
    | PROCEDURE-PARAMETERS = <text 0..1800 with-low>
    |
    | ESCAPE-CHARACTER = *NONE / <c-string 1..1>
,INTERRUPTION-ALLOWED = *YES / *NO

```

Operands

LOGGING =

Type of logging used for the procedure run. SDF statements are treated like commands for the purpose of logging.

LOGGING = *NO

No log of the procedure run is written to SYSOUT.

LOGGING = *ALL

Commands and input data for the procedure file are logged on SYSOUT when the procedure is executed.

LOGGING = *DATA

The input data for the procedure file is logged on SYSOUT when the procedure is executed.

LOGGING = *CMD

Commands and SDF statements of the procedure file are logged on SYSOUT when the procedure is executed.

PARAMETERS = *NO / *YES(...)

Specifies whether the procedure contains symbolic parameters.

PARAMETERS = *YES(...)

PROCEDURE-PARAMETERS = <text 0..1800 with-low>

Names of the symbolic parameters to be used in the procedure file. They form the so-called parameter list for the procedure.

Permissible characters for symbolic parameters:

first character: &
second character: letter
subsequent letters and digits
characters:
minimum length: 2 characters
maximum length: 255 characters

The individual symbolic parameters must be separated by commas; the entire expression must be enclosed in parentheses.

Symbolic parameters may be defined as positional parameters or as keyword parameters.

Positional parameters consist of only the name, e.g. &A.

Keyword parameters consist of the name followed by an equals sign, e.g. &A=. In addition, the name may be assigned a value, e.g. &A=VALUE. This value is used if the operand concerned is omitted when the procedure is called (CALL-PROCEDURE). The value must not exceed 254 characters in length.

Strings enclosed in apostrophes are also permitted as a value. The value must be enclosed in apostrophes if it contains blanks or special characters. Apostrophes within strings must be doubled. Lowercase letters within strings are retained.

If values are entered during procedure execution (prompting), lowercase letters will be converted into uppercase characters also within strings. The slash preceding a command cannot be replaced by parameters. Keyword parameters and positional parameters may appear in any order in the parameter list. If a symbolic parameter is followed in the procedure file by a period or an alphanumeric character, the parameter must be separated from it by a period.

ESCAPE-CHARACTER = *NONE / <c-string 1..1>

Character (&, @, #, \$ or *) identifying symbolic parameters in input data. If no character is specified (= presetting NONE) symbolic parameters in input data are not replaced. SDF statements are treated as commands, i.e. symbolic parameters must begin with & there and are always replaced. Symbolic parameters in input data must begin with the specified character. If the character is duplicated, it is replaced by a single character preventing parameter replacement.

INTERRUPTION-ALLOWED =

Defines whether the procedure can be interrupted with the K2 key for command input. Commands within the procedure which result in an interrupt are executed nonetheless.

INTERRUPTION-ALLOWED = *YES

Interrupting the procedure with the K2 key is possible.

The user may enter commands.

The specification is ignored if the procedure is called from a non-interruptible procedure.

INTERRUPTION-ALLOWED = *NO

Interruption of the procedure via the K2 key is not possible. Nested procedures for subsequent procedure levels are also not interruptible (regardless of the setting in INTERRUPTION-ALLOWED). If a program in which the event class ESCPBRK has been defined is called in a procedure, this STXIT routine is always activated by means of the K2 key.

Return codes

The BEGIN-PROCEDURE command is not interpreted by SDF. It can only result in an error if used in an incorrect context. In such a case the following command return code is issued.

(SC2)	SC1	Maincode	Meaning
	64	SSM2057	Command not the first command of a non-S procedure

Notes

- The BEGIN-PROCEDURE command must always be the first record in a procedure file. This file must not contain any further BEGIN-PROCEDURE commands or LOGON or RESTART-PROGRAM commands.
- The description of the CALL-PROCEDURE command explains how a procedure file is called and how the symbolic parameters are replaced with current values.
- The description of the END-PROCEDURE command explains how a procedure file is terminated.

Examples

Example 1

The procedure file MJ.PROC begins with the following record:

```
/BEG-PROC LOG=*ALL,PAR=*YES(PROC-PAR=(&A,&B),ESC-CHAR=C'#')
```

By means of the procedure call:

```
/CALL-PROC FROM-FILE=MJ.PROC,PROC-PAR=(MODULE,LISTE)
```

the symbolic parameters are given values and the procedure is executed.

With the call:

```
/CALL-PROC FROM-FILE=MJ.PROC
```

the values are not defined until during the procedure run (prompting).

Example 2

The procedure file MJ.PROC contains the following records:

```
/BEG-PROC LOG=*ALL,PAR=*YES(PROC-PAR=(&A,&B),ESC-CHAR=C'#')  
/MODIFY-JOB-SWITCHES ON=(1,4,5)  
/ASS-SYSDTA TO=*SYSCMD  
/START-EXE FROM-FILE=$LMS  
LIB MJ.BIBLIO.#A,BOTH,NEW  
ADDJ #B  
END  
/SHOW-FILE-ATTR F-NAME=MJ.BIBLIO.&A,INF=*ALL-ATTR  
/MODIFY-JOB-SWITCHES OFF=(1,4,5)  
/END-PROC
```

By means of the procedure call:

```
/CALL-PROC FROM-FILE=MJ.PROC,PROC-PAR=(MODULE,LISTE)
```

the following commands and LMS statements are executed:

```
...  
/ASS-SYSDTA TO=*SYSCMD  
/START-EXE FROM-FILE=$LMS  
LIB MJ.BIBLIO.MODULE,BOTH,NEW  
ADDJ LISTE  
END  
/SHOW-FILE-ATTR F-NAME=MJ.BIBLIO.MODULE,INF=*ALL-ATTR  
...
```


Example 3

The procedure file MJ.PROC1 contains the following records:

```
/BEG-PROC LOG=*ALL,PAR=*YES( PROC-PAR=( &IN, &OUT=MJ.OUT ), ESC-CHAR=C'&' )
/ASS-SYSDTA TO=*SYSCMD
/MOD-JOB-SW ON=(4,5)
/START-EDT
@READ'&IN'
...

@WRITE'&OUT'
@HALT
/MOD-JOB-SW OFF=(4,5)
/END-PROC
```

When called with:

```
/CALL-PROC FROM-FILE=MJ.PROC1
```

the following output appears on the screen:

```
%/BEG-PROC LOG=*ALL,PAR=*YES( PROC-PAR=( &IN, &OUT=MJ.OUT ), ESC-CHAR=C'&' )
%/ASS-SYSDTA TO=*SYSCMD
%/MOD-JOB-SW ON=(4,5)
%/START-EDT
%@READ'&IN'
&IN=
mj.inputfile
%@READ'MJ.INPUTFILE'
...

%@WRITE'MJ.OUT'
%@HALT
%/MOD-JOB-SW OFF=(4,5)
%/END-PROC
```

Thus, the symbolic parameter &OUT is already preset by the procedure declaration; only &IN remains to be specified during the procedure run.

BROADCAST

Send message to all active user tasks

Description status:	BS2000 OSD/BC V10.0A
Functional area:	not allocated
Domain:	not allocated
Privileges:	OPERATING TSOS
Routing code:	E



The BROADCAST command is being replaced by the INFORM-ALL-JOBS command.

BROADCAST continues to be supported to ensure backwards compatibility. For new applications INFORM-ALL-JOBS should be used.

All that follows is a brief overview of the command's function, syntax and return codes.

Function

The BROADCAST command sends a message from the operator to all interactive users currently connected to the system. The system adds the date and time of day to the operator's message.

Format

BROADCAST	Alias: BCST
MSG = <text 1..72>	

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
2	0	NBR0725	Warning: The message could not be sent to all interactive tasks
2	32	NBR0727	Internal error: Error in \$PSTMS call (incorrect UNIT, FUNCTION or VERSION)
	64	CMD0216	User does not have the required privilege
	130	EXC0061	Class 4 memory request failed; command processing aborted

CALL-PROCEDURE

Run command sequence from procedure file or list variable

Description status:	SYSFILE V19.0A
Functional area:	Procedures
Domain:	PROCEDURE
Privileges:	STD-PROCESSING OPERATING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

The CALL-PROCEDURE command starts a stored command sequence (procedure). During processing, symbolic parameters contained in the sequence are replaced by the values specified in the command call (PROCEDURE-PARAMETERS operand). A procedure may contain the following components:

- commands (starting with a slash)
- SDF statements, i.e. statements for a program which are defined in a syntax file (starting with a double slash)
- input data, i.e. data, parameters and statements read by a program

During invocation, the procedure is assigned to the system file SYSCMD as the input source for commands. If the procedure includes SDF statements and/or input data, the SYSDTA system file must be assigned to the SYSCMD system file (see ASSIGN-SYSDTA, or SET-PROCEDURE-OPTIONS for S procedures).

Procedures can be stored as:

- a cataloged SAM or ISAM file (even a temporary one) with records of variable length
- a type J or SYSJ element in a PLAM library
- an S variable of the “list” type

There are two types of procedure, which differ in formal structure and operational sequence:

- non-S procedure
- S procedure

Restrictions

Users with SECURITY-ADMINISTRATION, SAT-FILE-EVALUATION or SAT-FILE-MANAGEMENT privilege can use the command in procedures only.

A procedure cannot be started from an S variable by specifying FROM=*VARIABLE(...) unless the chargeable SDF-P subsystem is loaded.

non-S procedure

The first data record in the non-S procedure must be the command BEGIN-PROCEDURE. There symbolic parameters, logging and interruptibility of the procedure can be specified. Next come the commands, SDF statements and input data to be processed. Processing ends with the command END-, CANCEL- or EXIT-PROCEDURE or the last data record. If an error occurs, spin-off is activated. Error handling is described under the SET-JOB-STEP command.

S procedure

The first record in an S procedure must be a command. The commands BEGIN-PROCEDURE, or PROCEDURE (ISP command), and SET-LOGON-PARAMETERS, or LOGON (ISP command), are not allowed. Procedure attributes can be explicitly specified via SET-PROCEDURE-OPTIONS as the first command. If no explicit specification is made, the SDF-P presettings apply implicitly (see SET-PROCEDURE-OPTIONS). Once the procedure attributes have been defined, the DECLARE-PARAMETER command can be used to define symbolic parameters, which are replaced by the values of the like-named S variables during execution. S procedures can only execute if S variables are available, i.e. if at least the SDFPBASY subsystem (see [section “SDF-P-BASYS” on page 1-131](#) and the “SDF-P” manual [34]) is loaded.

Processing of commands, SDF statements and input data ends with the CANCEL- or EXIT-PROCEDURE command or with the final record. If an error occurs, SDF-P error recovery is activated. Error recovery is described under the command IF-BLOCK-ERROR.

Two different procedure formats are supported:

- text procedure The S procedure is in its original text format. The full range of SDF-P functions is available only if the chargeable subsystem SDF-P is loaded when the procedure is called. In libraries, element type J should be used for text procedures.
- object procedure An S procedure in text format has been translated to object format with the COMPILE-PROCEDURE command. The COMPILE-PROCEDURE command is part of the chargeable SDF-P subsystem. The object procedure can utilize the full functionality of SDF-P (apart from the COMPILE-PROCEDURE command) irrespective of whether the SDF-P subsystem is currently available or not. In libraries, element type SYSJ (the default for COMPILE-PROCEDURE) should be used for object procedures.

Automatic procedure start

LOGON procedure

After LOGON processing, SDF automatically starts the system LOGON procedure followed by the user LOGON procedure. A system LOGON procedure is started when system administration makes one available. A user LOGON procedure is started as a call procedure if it is cataloged in the relevant user identification under the name `SYS.SDF.LOGON.USERPROC` (if it is under `SYS.SDF.LOGON.USERINCL` it is called as an include procedure). LOGON procedures are ignored in the following cases and no warning is issued:

- The procedure file is simply catalogued but does not occupy any storage space.
- The task is an RFA task.
- The task does not possess any privileges apart from `HARDWARE-MAINTENANCE`, `SECURITY-ADMINISTRATION`, `SAT-FILE-MANAGEMENT` and `SAT-FILE-EVALUATION`.

Input is not possible until the LOGON procedures have been executed.

LOGOFF procedure

During LOGOFF processing, SDF automatically starts the system procedure and then the user LOGOFF procedure. A system LOGOFF procedure is started if one has been made available by systems support (entry in the SDF parameter file). A user LOGOFF procedure is started as a call procedure if it is catalogued under the name `SYS.SDF.LOGOFF.USERPROC` in the associated user logon (under `SYS.SDF.LOGOFF.USERINCL` the call takes the form of a call procedure). LOGOFF procedures are ignored in the following cases and no warning is issued:

- The procedure file is simply catalogued but does not occupy any storage space.
- The task was cancelled with `CANCEL-JOB` or `FORCE-JOB-CANCEL`.
- The task is an RFA task.
- The task does not possess any privileges apart from `HARDWARE-MAINTENANCE`, `SECURITY-ADMINISTRATION`, `SAT-FILE-MANAGEMENT` and `SAT-FILE-EVALUATION`.

Implemented procedure

If commands implemented as procedures occur in an active syntax file (see the “SDF-A” manual [33]), the specified procedure is started when such a command is called.

Nesting of procedures

Procedures may be nested to any depth: If the command sequence of a procedure file contains a `CALL-PROCEDURE` command, the sequence is interrupted and the next procedure is started, etc.

CALL-PROCEDURE

At the end of a procedure (END-PROCEDURE, EXIT-PROCEDURE or EOF condition), control branches back to the interrupt point of the procedure last exited. The figure below provides an illustration of procedure nesting with CALL-PROCEDURE for a *non-S procedure*.

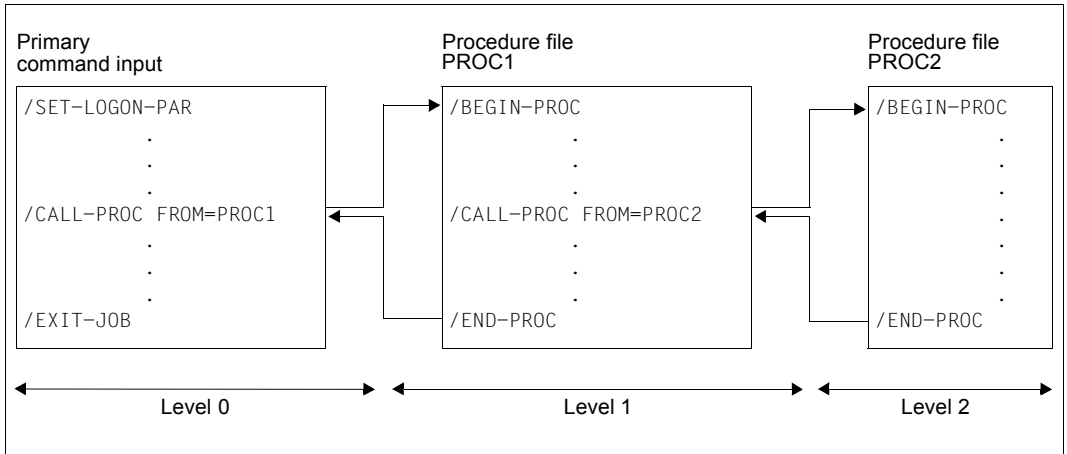


Figure 1: CALL-PROCEDURE procedure nesting

Format

CALL-PROCEDURE	Alias: CL / CLP
<p>FROM-FILE = <filename 1..54 without-gen> / *LIBRARY-ELEMENT(...) / *VARIABLE(...)</p> <p>*LIBRARY-ELEMENT(...)</p> <ul style="list-style-type: none"> LIBRARY = <filename 1..54 without-gen> ,ELEMENT = <composed-name 1..64>(…) <composed-name 1..64>(…) VERSION = *HIGHEST-EXISTING / <composed-name 1..24> ,TYPE = *STD / *BY-LATEST-MODIFICATION / <alphanum-name 1..8> <p>*VARIABLE(...)</p> <ul style="list-style-type: none"> VARIABLE-NAME = <composed-name 1..255> <p>,PROCEDURE-PARAMETERS = *NO / <text 0..1800 with-low></p> <p>,LOGGING = *PARAMETERS(...) / YES / *NO /</p> <p>*PARAMETERS(...)</p> <ul style="list-style-type: none"> CMD = *BY-PROC-TEST-OPTION / *YES / *NO ,DATA = *BY-PROC-TEST-OPTION / *YES / *NO <p>,UNLOAD-ALLOWED = *YES / *NO</p> <p>,EXECUTION = *YES / *NO</p>	

Operands

FROM-FILE = <filename 1..54 without-gen> / ***LIBRARY-ELEMENT**(...) / ***VARIABLE**(...)

Name of the procedure file.



If the file/library name is specified without a catalog/user ID and if it is not cataloged in the user ID, the system tries to access a file or library of the same name in the system default ID. (For information on this “secondary read” function see the “Introductory Guide to DMS” [13].)

FROM-FILE = ***LIBRARY-ELEMENT**(...)

The procedure is stored in a PLAM library element.

LIBRARY = <filename 1..54 without-gen>

Name of the PLAM library containing the procedure as an element (type J or SYSJ: see TYPE operand).

ELEMENT = <composed-name 1..64>(…)

Name of the element.

VERSION = *HIGHEST-EXISTING / <composed-name 1..24>

Version of the library element. The default value is HIGHEST-EXISTING, i.e. the procedure is taken from the element with the highest version.

TYPE = *STD / *BY-LATEST-MODIFICATION / <alphanum-name 1..8>

Designates the element type the procedure is stored under in the PLAM library.

TYPE = *STD

The procedure can be stored as an element of type SYSJ or J. The specified element is first searched for among the type SYSJ elements. If it is not found there, the search proceeds to the type J elements.

A non-S procedure can only be a type J element. An S procedure may be either a text procedure (original text format) or an object procedure (compiled object format). To simplify maintenance of the two formats in a library, text procedures should be stored as type J elements, object procedures as type SYSJ elements. The COMPILE-PROCEDURE command (part of the chargeable SDF-P subsystem) by default generates an object procedure of type SJ from a text procedure of type J. If this convention is followed, specifying TYPE=*STD (the default value) ensures that object procedures will be given precedence over text procedures.

TYPE = *BY-LATEST-MODIFICATION

The procedure can be stored as an element of type SYSJ or J. If the specified element exists both as type SYSJ and as type J, the element most recently modified will be called. If the time stamp is identical, the type SYSJ element will be called. Specifying TYPE=*BY-LATEST-MODIFICATION ensures that the most up-to-date element will be called, typically during the debugging phase when a procedure is being written or modified.

TYPE = <alphanum-name 1..8>

The procedure will be searched among elements of the specified type only.

FROM-FILE = *VARIABLE(…)

Only possible if the chargeable software product SDF-P is loaded The procedure is stored in an S variable of the “list” type.

VARIABLE-NAME = <composed-name 1..255>

Name of the S variable.

PROCEDURE-PARAMETERS = *NO / <text 0..1800 with-low>

Parameter values to be used in place of the corresponding symbolic parameters in the procedure file. For “text”, the following format applies:

$$([\dots]) \left\{ \begin{array}{l} \text{sympar=} \\ \text{paramwert} \\ \text{sympar=paramwert} \end{array} \right\}$$

- *sympar* is the name of a symbolic keyword parameter declared either in the BEGIN-PROCEDURE command (without the initial “&”) in the non-S procedure file, or in the DECLARE-PARAMETER command of the S procedure file.
- *paramval* is the current value either of a keyword or positional parameter declared in the BEGIN-PROCEDURE command (maximum length 254 characters), or of a procedure parameter defined in the DECLARE-PARAMETER command. Character strings enclosed in apostrophes are also allowed as actual parameter values. The value must be enclosed in apostrophes if it contains blanks or special characters. Apostrophes within strings must be doubled. Lowercase letters within strings are retained.

The null string is interpreted as a missing parameter value and, in interactive mode, leads to a request for a value during procedure execution. Values input during procedure execution (prompting) are converted to uppercase letters also within strings. If the current value of a parameter is requested during procedure execution and the user presses the K2 key, the system issues message SSM2060 asking if the user wants to terminate the procedure. If the procedure is continued, the value is queried again.

The current “paramval” parameter values from the CALL-PROCEDURE command replace the symbolic parameters as follows:

In the case of S procedures, parameters can be transferred as keyword or positional parameters since no specification in this regard is made in the DECLARE-PARAMETER command. However, a positional parameter must not follow a keyword parameter. Prompting is only possible if it has been defined (INITIAL-VALUE=*PROMPT).

In the case of non-S procedures, parameters can only be transferred as defined in the BEGIN-PROCEDURE command, i.e. positional parameters if defined as such and keyword parameters if defined as such.

- **Keyword parameters** receive their current values from the parameter list of the CALL-PROCEDURE command. If a value is not specified, the value defined in the procedure is used. If no value was defined there either, the value is requested, in interactive mode, when the procedure is executed. This is called “prompting”.

The following table provides an overview of related parameter specifications in the BEGIN-PROCEDURE and CALL-PROCEDURE commands:

		Specification in BEGIN-PROCEDURE	
		&X=ABC	&X=
Specification in CALL-PROCEDURE	X=	Prompting	Prompting
	X=DEF	DEF	DEF
	without	ABC	Prompting

- **Positional parameters** are assigned, in sequence, those current parameter values from the CALL-PROCEDURE command that are specified without the keyword “sympar=”. If the current value of a positional parameter is not specified in the CALL-PROCEDURE command (a comma stands for the value), it will be requested in interactive mode (prompting) when the procedure is executed (if the value is required at all).

LOGGING = *PARAMETERS(...) / *YES / *NO

This controls logging of procedure execution.

The LOGGING operand is ignored when *non-S procedures* are called, since in this case logging can only be declared in the procedure head (see the LOGGING operand in the BEGIN-PROCEDURE command). When an S procedure is logged, every procedure line that is processed is output with the line number and procedure level prefixed to it.

LOGGING = *PARAMETERS(...)

Logging can be set separately for command/statement lines and for data lines.

CMD = *BY-PROC-TEST-OPTION / *YES / *NO

This specifies whether commands are to be logged. The default value is BY-PROC-TEST-OPTION, i.e. no logging (equivalent to *NO) or the value selected as the default by the user with the MODIFY-PROC-TEST-OPTIONS command (component of the chargeable SDF-P subsystem).

DATA = *BY-PROC-TEST-OPTION / *YES / *NO

This specifies whether data lines are to be logged. The default value is BY-PROC-TEST-OPTION, i.e. no logging (equivalent to *NO) or the value selected as the default by the user with the MODIFY-PROC-TEST-OPTIONS command (component of the chargeable SDF-P subsystem).

UNLOAD-ALLOWED = *YES / *NO

This specifies whether a program that was loaded when the procedure was called may be unloaded. Protection against unloading is guaranteed *only* for unloading by means of the commands START-(EXECUTABLE-)PROGRAM, LOAD-(EXECUTABLE-)PROGRAM and CANCEL-PROGRAM.

The specification YES is ignored if the procedure is called from a procedure for which UNLOAD-ALLOWED=*NO was declared.

EXECUTION = *YES / *NO

This specifies whether the procedure is merely to be analyzed for test purposes or whether it is also to be executed. Only EXECUTION=*YES may be specified for *non-S procedures*.

Testing is possible via the MODE operand of the MODIFY-SDF-OPTIONS command.

Return codes

The following command return codes can only be returned if the called procedure does not supply any command return code itself (e.g. EXIT-PROCEDURE not executed). Command return codes whose maincode begins with “SSM” can only be returned when a non-S procedure is called. Command return codes whose maincode begins with “SDP” can only be returned when an S procedure is called.

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
2	0	SSM2058	Protocol type error
2	0	SSM2065	EOF on procedure file, /END-PROC simulated
	1	SSM2036	Incomplete operand
	1	SSM2054	Symbolic operand error
	1	SSM2055	Symbolic operand error in /BEGIN-PROC
	1	SDP0138	Error during pre-analysis of the text procedure or object procedure faulty
	1	CMD0202	Syntax error
	3	CMD2203	Incorrect syntax file
	32	CMD0221	System error (internal error)
	64	SDP0093	Non-S procedure can only be type J element.
	64	SDP0144	Error during parameter transfer
	64	SSM2052	DMS error (OPEN error)
	64	SSM2053	Not a SAM/ISAM file or file does not begin with /BEGIN-PROC or /PROC
	64	SSM2056	Parameters of /CALL-PROC and /BEGIN-PROC incompatible
	64	SSM2061	Error on accessing library element
	64	SSM2064	Procedure file cannot be fetched from remote system
	130	SDP0099	No more address space available
xx	xx	xxxxxxx	Other return codes of the called procedure

Notes

- Symbolic operands may be used anywhere in any command in the procedure file. However, they cannot replace the initial slash preceding a command.
- When the procedure records processed are logged to SYSOUT, symbolic operands are replaced with the current operand values.
- Procedure files may be protected with passwords against reading, overwriting and execution, e.g. using the MODIFY-FILE-ATTRIBUTES command. The execute password or a higher-ranking password must be given in an ADD-PASSWORD command before a CALL-PROCEDURE command is issued.

- When a library element is used, a temporary SAM file named S.IN.library.element.tsn.hhmmss.nnnn is created (only for non-S procedures), which contains the element. where:

library	Library name (up to 20 characters are printed)
tsn	Task sequence number
hhmmss	Time in hours-minutes-seconds
nnnn	Sequence number

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

- EXIT-JOB or LOGOFF in the ENTER file
- END-PROCEDURE in the procedure
- If the CALL-PROCEDURE command is called via the CMD macro, the calling program is unloaded. Any ABEND-STXIT routine defined in the program is not activated. For details of the CMD macro, ABEND-STXIT routine, refer to the “Executive Macros” manual [22].
- If a non-S procedure file contains statement or data records in addition to command records, the following command record must be written in the procedure file prior to the statement or data records:

```
/ASSIGN-SYSDTA TO=*SYSCMD
```

This assignment need not be repeated in the event of nesting. In S procedures, this depends on the setting in the SET-PROCEDURE-OPTIONS command.

CANCEL-CCOPY-SESSION

Cancel CCOPY-Session

Description status:	CCOPY V9.0B
Functional area:	Pubset and MRSCAT management
Domain:	STORAGE-MANAGEMENT
Privileges:	TSOS HSMS-ADMINISTRATION

Function

The CANCEL-CCOPY-SESSION command cancels an active CCOPY session.

The user can have all CCOPY sessions displayed using the SHOW-CCOPY-SESSION command.

Format

CANCEL-CCOPY-SESSION

SESSION-ID = <alphanum-name 8..8>

Operands

SESSION-ID = <alphanum-name 8..8>

Identification (eight positions) of the CCOPY session that is to be canceled.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
	32	DCH000A	Internal error
	32	DCH000B	Unexpected return code when a CCOPY session was canceled
	32	DCH000D	System error
	64	CMD0216	Privileges error
	64	DCH0007	Invalid session ID
	64	DCH0008	CCOPY session already terminated abnormally
	64	DCH0009	CCOPY session already terminated normally

CANCEL-JOB

Cancel user job

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Job processing
Domain:	JOB
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE OPERATING SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION
Routing code:	P

Function

The CANCEL-JOB command cancels jobs running under the user's own ID. The command can also be used to cancel batch jobs generated by the user's own ID and running under a different user ID (extended access).

Resources occupied by the job are released (as for EXIT-JOB or LOGOFF). A job cannot use this command to terminate itself (only possible with EXIT-JOB or LOGOFF). If the job to be canceled has already been terminated, the CANCEL-JOB command is rejected and a message to this effect is displayed at the terminal. The CANCEL-JOB command cancels a job only when the job is in TU status or when processing of the next command is imminent. A job cannot be canceled in the following circumstances:

- The job is waiting for an operator response. The command takes effect as soon as there has been a response (see the FORCE-JOB-CANCEL command to find out how to cancel jobs without waiting for a response).
- The job terminates abnormally due to a system error
- A HOLD-TASK command was issued for the job (by systems support personnel)
- The job is already in job termination; e.g.: actions following EXIT-JOB or LOGOFF, or actions in the user program after ABEND has been signaled (for more on the ABEND-STXIT routine, see the “Executive Macros” manual [22].)
- The job is in “PENDED INDEFINITELY” state
- The job is a system task.

If a job is to be canceled in interactive mode with CANCEL-JOB, the system requests confirmation if the job to be canceled was started with the attribute PROTECTION=*CANCEL (see ENTER-JOB, ENTER-PROCEDURE or SET-LOGON-PARAMETERS command).

Information on job termination

In the case of a job canceled with CANCEL-JOB, information about the originator of the CANCEL-JOB command is output to SYSOUT for that job. The output to SYSOUT contains an additional comment specified in the TEXT operand of the CANCEL-JOB command. If the job was terminated abnormally, i.e. the CANCEL-JOB command was issued with STEPS=*ALL (default), the status indicator of a monitoring job variable is set to the value '\$A'. In addition, the information about the originator in abbreviated form and the comment from the TEXT operand are included in the system part (the first 128 bytes) of the monitoring job variable:

- The originator information starts at byte 37 and consists of the character string *CAN:'info'*, where *info* contains the first 27 bytes of the originator information supplied to SYSOUT.
- The comment starts at byte 70 and consists of the character string *TEXT:'text'*, where *text* contains the first 51 bytes of the comment specified in the operand.

For information on job monitoring, see the “Job Variables” manual [20].

Privileged functions

If the CANCEL-JOB command is issued at the operator terminal or under a user ID with TSOS or OPERATING privilege, it can be used to cancel a job issued under any user ID.

Format

CANCEL-JOB	Alias: CNJ
<p>JOB-IDENTIFICATION = *TSN(...) / *MONJV(...)</p> <p> *TSN(...)</p> <p> TSN = <alphanum-name 1..4></p> <p> HOST = *STD / <c-string 1..8></p> <p> *MONJV(...)</p> <p> MONJV = <filename 1..54 without-gen-vers></p> <p>,DUMP = *NO / *STD / *CANCEL-RUNNING-DUMP</p> <p>,SYSTEM-FILES = *STDOUT / *PRINT / *MAIL / *DELETE</p> <p>,STEPS = *ALL-STEPS / *CURRENT-STEP / *ALL-CALENDAR-REPETITIONS</p> <p>,TEXT = *NO / <c-string 1..72></p>	

Operands

JOB-IDENTIFICATION =

Type of job identification.

JOB-IDENTIFICATION = *TSN(...)

The job is identified by its task serial number (TSN).

TSN = <alphanum-name 1..4>

TSN of the job to be canceled.

Leading zeros may be omitted

HOST = *STD / <c-string 1..8>

Host system on which the job is running. The default is *STD, which means that the job is running on the local host. The host name of a remote system can only be specified for systems which are on a computer network (see the "HIPLEX MSCF" manual [25]).

JOB-IDENTIFICATION = *MONJV(...)

The job is identified by means of the monitoring job variable. The command is rejected if the JV is not accessible (no read permission or non-existent JV) or is not monitoring a job.

For a job running on a remote host to be accessible via the monitoring JV, the MRSCAT of each host must contain the catalog ID of the pubset of the other host.

MONJV = <filename 1..54 without-gen-vers>

Name of the JV monitoring the job to be canceled.

The job to which this job variable is assigned is then canceled. This operand is available only to users who have the software product JV.

DUMP =

Defines whether a user dump is to be output for the job to be canceled or whether a user or area dump that is currently being processed is to be cancelled.

DUMP = *NO

No memory dump.

DUMP = *STD

The definition currently made in the job that is to be canceled (see DUMP operand in the MODIFY-TEST-OPTIONS command) is interpreted as follows:

Specification for the job that is to be canceled	Effects
DUMP=*NO	No memory dump.
DUMP=*STD / *YES / *SYSTEM	A user dump is requested, for DUMP=*SYSTEM eventually a system dump is requested.

DUMP = *CANCEL-RUNNING-DUMP

If a user or area dump is currently being created for the job that is to be cancelled then dump creation should be cancelled immediately and the dump file deleted.

SYSTEM-FILES =

Specifies whether the system files are to be output to printer or sent by email. Output to printer is not possible if the value NO-SPOOL was set for the system parameter SSMLGOF1.

SYSTEM-FILES = *STDOUT

Depending on the setting of the system parameter SSMOUT, the system files are output to printer (see *PRINT) or sent by email (see *MAIL). In the case of cross-system command processing, the system parameter of the target system is evaluated.

SYSTEM-FILES = *PRINT

The system files are output to printer.

SYSTEM-FILES = *MAIL

The system files are sent by email. The receiver address(es) is/are taken over from the user entry of the user ID of the aborted job. How the receiver address is selected from an address list in accordance with the job name is described under the MAIL-FILE command. If transfer by email is not possible (e.g. no email address in the user ID), the system files are output to printer.

SYSTEM-FILES = *DELETE

Output of the system files is suppressed.

STEPS =

Specifies how the job is to be canceled. Information about the originator of the CANCEL-JOB command is always output to SYSOUT for a job that is to be canceled. The information output is supplemented by the comment specified by the originator in the TEXT operand of the CANCEL-JOB command.

STEPS = *ALL-STEPS

The specified job is canceled in its entirety. The status indicator of a monitoring job variable is set to '\$A'. The information about the originator of the CANCEL-JOB command is included in abbreviated form in the system part (the first 128 bytes) of the monitoring job variable. In addition, a text specified in the TEXT operand is included in the system part of the monitoring job variable.

The effect depends on the batch job type of the specified job:

Repeat job: Only the cancellation of the Type 1 follow-up job of a repeat job will result in the termination of all future job repetitions.

Calendar job: Cancellation of a calendar job affects only the current job which is still on the job queue or is executing. It does not affect subsequent repetitions of the calendar job. STEPS=*ALL-CALENDAR-REPETITIONS can be used to terminate the entire calendar job.

STEPS = *CURRENT-STEP

– The job is being processed:

In the specified job, only the current processing step is canceled: the spin-off mechanism or SDF-P error handling is triggered. No information is included in a monitoring job variable. The job can be terminated normally if no other reason for abnormal termination occurs. If a user program is currently loaded, it is first terminated abnormally. If an error handling facility is provided within the ENTER file (for method, see SET-JOB-STEP command), then the job is continued at the next processing step. If the next processing step is termination of the job (EXIT-JOB or LOGOFF command), the job ends normally and a monitoring job variable is then set to '\$T' (normal end of job).

Note

On terminating a job with CANCEL-JOB and STEPS=*CURRENT-STEP, no abnormal end of job results.

- The job is waiting to be processed:
The job is canceled in its entirety as with STEPS=*ALL-STEPS (see above).

STEPS = *ALL-CALENDAR-REPETITIONS

The specified job is canceled as with STEPS=*ALL-STEPS. If it is a calendar job, the entire calendar job (i.e. including any future repetitions) is removed from the job management system.

TEXT = *NO / <c-string 1..72>

Specifies whether and if so which text is to be output to SYSOUT as a comment in the specified job before abnormal termination. The text can be up to 72 characters long. If job monitoring is in use and the value *ALL-STEPS (default) or *ALL-CALENDAR-REPETITIONS was specified in the STEPS operand, the first 51 characters of the specified text are also included in the system part of the monitoring job variable.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
2	0	CMD0002	Command executed with a warning
	32	CMD0221	System error
	64	JMS0630	Semantic error
	64	JMS0640	MONJV error, illegal access or incorrect call time
	64	JMS0670	Error in a REMOTE job
	130	JMS0620	Memory saturation
	130	JMS0650	MSCF or target computer not available
	130	JMS0660	Retry command later

If the command is issued for a print job, command return codes are returned by SPOOL (see also CANCEL-PRINT-JOB).

Notes

- The CANCEL-JOB may be issued in batch or interactive mode or at the console.
- With spoolout tasks, more than one may have the same task sequence number (TSN), as a result of a PRINT-DOCUMENT command for several files, for example. In that case, the CANCEL-JOB command applies to all print jobs which have the TSN specified in the command. A message is issued for each job that is canceled. The SHOW-USER-STATUS command outputs a list of jobs that have not yet been canceled. The jobs in question may be undergoing processing or may still be waiting for processing.

Notes on job monitoring (see also the “Job Variables” manual [20]):

- If a job being monitored by a job variable (JV) is canceled, the JV job variable is set to “\$A” provided that STEPS=*ALL-STEPS or *ALL-CALENDAR-REPETITIONS has been set. With a waiting job this also applies to STEP=*CURRENT-STEP. If a program executing within the job is being monitored by a job variable, the status indicator of the job variable monitoring the program is likewise set to “\$A”.
- If the job variable is not accessible or is not being used to monitor a job, the command is rejected.
- If a calendar job is being monitored, the job variable doing the monitoring is assigned to the calendar job for as long as it exists. Monitoring ends only when the entire calendar job ends.

Example

*Job cancellation with STEPS=*CURRENT-STEP and STEPS=*ALL-STEPS*

```

/enter-proc proc.sort1,proc-par=(monjv-1='JV.JOB-A',
                                output-file-1='OUT.WORK1',
                                monjv-2='JV.JOB-B',
                                output-file-2='OUT.WORK2'),
                                monjv=jv.job-c,job-class=jcb00200,job-name=jobc,
                                logging=*yes,list=*yes,
                                ass-sys-file=*par( syslst=lst.jobc, sysout=out.jobc) _____ (1)
% JMS0066 JOB 'JOB' ACCEPTED ON 12-01-26 AT 15:17, TSN = 3182
/show-job-status monjv(jv.job-c) _____ (2)
%TSN:      9VH1      TYPE:      2 BATCH      NOW:      2012-01-26.151748
%JOBNAME:  JOB      PRI:      9 225      SPOOLIN: 2012-01-26.1517
%USERID:   USER1    JCLASS:   JCB00200     LOGON:   2012-01-26.1517
%ACCNB:    89001    CPU-MAX:   200      CPU-USED:000000.1831
%REPEAT:   NO      RERUN:    NO      FLUSH:   NO
%MRSCAT:   HOLD:   NO      START:   SOON
%TID:      003A0193 UNP/Q#:   17/012
%CMD:      WAIT
%ORIGFILE: :20SG:$USER1.PROC.SORT1
%CMD-FILE: :20SG:$USER1.S.E.9VC3.2012-01-26.15.17.37
%MONJV:    :20SG:$USER1.JV.JOB-C
/show-job-status monjv(jv.job-a) _____ (3)
% JVS04D1 MONITORING JOB VARIABLE ':20SG:$USER1.JV.JOB-A' NOT ASSIGNED TO JOB
/cancel-job monjv(jv.job-c),steps=*current,
  text='Auftrag JOB-A wurde nicht gestartet; deshalb Abbruch von WORK-1' _____ (4)
% CAN000K CANCEL PROCESSING STARTED FOR TSN '3182' WITH USER ID 'USER1'
/show-job-status monjv(jv.job-c) _____ (5)
%TSN:      9VH1      TYPE:      2 BATCH      NOW:      2012-01-26.151943
%JOBNAME:  JOB      PRI:      9 225      SPOOLIN: 2012-01-26.1517
%USERID:   USER1    JCLASS:   JCB00200     LOGON:   2012-01-26.1517
%ACCNB:    89001    CPU-MAX:   200      CPU-USED:000000.1942
%REPEAT:   NO      RERUN:    NO      FLUSH:   NO
%MRSCAT:   HOLD:   NO      START:   SOON
%TID:      003A0193 UNP/Q#:   17/012
%CMD:      WAIT
%ORIGFILE: :20SG:$USER1.PROC.SORT1
%CMD-FILE: :20SG:$USER1.S.E.9VC3.2012-01-26.15.17.37
%MONJV:    :20SG:$USER1.JV.JOB-C
/show-jv jv=jv.job-c _____ (6)
%$R 09VH110SH J0312012-01-26141737

/show-job-status monjv(jv.job-b) _____ (7)
% JVS04D1 MONITORING JOB VARIABLE ':20SG:$USER1.JV.JOB-B' NOT ASSIGNED TO JOB
/cancel-job monjv(jv.job-c),steps=*all,text='Auch JOB-B nicht gestartet! Deshalb
  soll JOB-C abnormal beendet werden!' _____ (8)
% CAN000K CANCEL PROCESSING STARTED FOR TSN '3182' WITH USER ID 'USER1'
/show-jv jv=jv.job-c _____ (9)
%$A 09VH110SH J0312012-01-26141737CAN:'DIAL 9VC3 USER1 ULF ' TEXT:'AUC
H JOB-B NICHT GESTARTET! DESHALB SOLL JOB-C ABNO'

```

- (1) The S procedure *PROC.SORT1* is started as a batch job via the ENTER-PROCEDURE command. The procedure comprises two work steps:
 - WORK-1: Waiting for normal termination of a job monitored by the job variable identified by parameter *MONJV-1*. The name of the input file for a SORT run is passed in the user part of the job variable. The parameter *OUTPUT-FILE-1* determines the output file of the SORT run.
 - WORK-2: Waiting for normal termination of a job monitored by the job variable identified by parameter *MONJV-2*. The name of the input file for a SORT run is passed in the user part of the job variable. The parameter *OUTPUT-FILE-2* determines the output file of the SORT run.

Both work steps can be performed independently. The batch job is monitored using the job variable *JV.JOB-C*.

- (2) The started job *JOB-C* is waiting (WAIT-EVENT).
- (3) The first monitoring job variable *JV.JOB-A* is not assigned to any job.
- (4) So the section being processed is canceled with STEP=*CURRENT-STEP.
- (5) *JOB-C* is waiting again (second WAIT-EVENT).
- (6) The monitoring job variable of *JOB-C* has not been affected by CANCEL-JOB.
- (7) The second monitoring job variable *JV.JOB-B* is not assigned to any job either.
- (8) The entire *JOB-C* is not to be terminated normally, since further processing does not make sense.
- (9) The monitoring job variable indicates that the job has been abnormally terminated. It also contains the cause and the originator's comment in abbreviated form.

SYSOUT log with execution of JOB-C

```

/CALL-PROC NAME=:20SG:$USER1.S.PROC.9VC3.2012-01-26.15.17.37
PROCEDURE-PARAMETERS=(MONJV-1='jv.job-a',OUTPUT-FILE-1='out.work1',
MONJV-2='jv.job-b',OUTPUT-FILE-2='out.work2'),LOGGING=*YES
 1 1 /BEG-PAR-DECL
 2 1 /DECL-PAR MONJV-1 (INIT = *PROMPT)
 3 1 /DECL-PAR OUTPUT-FILE-1 (INIT = *PROMPT)
 4 1 /DECL-PAR MONJV-2 (INIT = *PROMPT)
 5 1 /DECL-PAR OUTPUT-FILE-2 (INIT = *PROMPT)
 6 1 /END-PAR-DECL
% SDP0116 PARENTHESIS MISSING
% SDP0014 WARNING IN LINE: 30 IN PROCEDURE ':20SG:$USER1.S.PROC.9VC3.2012-01-
26.15.17.37'
 7 1 /WORK-1:
 8 1 /WAIT-EVENT JV(COND = ( (jv.job-a,1,2) = '$T' ),
TIME-LIMIT = 3600 )
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:17:37
% CAN00BY CANCELLED BY 'DIAL 9VC3 USER1 ULF FIREBALL $$$06580'
% CANOTXT TEXT: 'AUFTRAG JOB-A WURDE NICHT GESTARTET; DESHALB ABRUCH VON WORK-1'
% SDP0004 ERROR DETECTED AT COMMAND LINE: 8 IN PROCEDURE
':20SG:$USER1.S.PROC.9VC3.2012-01-26.15.17.37'
 18 1 /WORK-1-ERROR:
 18 1 / IF-BLOCK-ERROR
 19 1 /WRITE-TEXT 'Fehler bei WORK-1 mit SC1 = 64'
Fehler bei WORK-1 mit SC1 = 64
 20 1 /HELP-MSG CMD0205
% CMD0205 ERROR IN PRECEDING COMMAND OR PROGRAM AND PROCEDURE STEP TERMINATION: COMMANDS
WILL BE IGNORED UNTIL /SET-JOB-STEP OR /LOGOFF OR /EXIT-JOB IS RECOGNIZED
% ? The command issued is invalid or the program was terminated with the
% macro TERM UNIT=STEP or TERMJ.
% All commands following the invalid one will be ignored until a
% /SET-JOB-STEP or a /LOGOFF or an /EXIT-JOB command is received.
% In a procedure, the /END-PROCEDURE command will be accepted but the
% other commands will still be ignored.
% RESPONSE : NONE
 23 1 /END-IF
 24 1 /WORK-2:
 25 1 /WAIT-EVENT JV(COND = ( (jv.job-b,1,2) = '$T' ),
TIME-LIMIT = 3600 )
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:19:28
% CAN00BY CANCELLED BY 'DIAL 9VC3 USER1 ULF FIREBALL $$$06580'
% CANOTXT TEXT: 'AUCH JOB-B NICHT GESTARTET! DESHALB SOLL JOB-C ABNORMAL BEENDET
WERDEN!'
% NRTT201 TASK TERMINATION DUE TO /CANCEL(-JOB) COMMAND
% EXC0419 /LOGOFF AT 1521 ON 12-01-26 FOR TSN '9VH1'
% EXC0421 CPU TIME USED: 0.1995

```

Contents of procedure file PROC.SORT1

```

/BEG-PAR-DECL
/  DECL-PAR  MONJV-1      (INIT = *PROMPT)
/  DECL-PAR  OUTPUT-FILE-1 (INIT = *PROMPT)
/  DECL-PAR  MONJV-2      (INIT = *PROMPT)
/  DECL-PAR  OUTPUT-FILE-2 (INIT = *PROMPT)
/END-PAR-DECL
/WORK-1:      "Sorting file 1"
/              WAIT-EVENT  JV(COND = ( (&(MONJV-1),1,2) = '$T' ),-
/                                  TIME-LIMIT = 3600 )
/              CRE-JV      JV = #WORK1
/              MOD-JV      JV = #WORK1, -
/                                  SET-VALUE = ( &(MONJV-1),129,54 )
/              ADD-FILE-LINK LINK=SORTIN, -
/                                  FILE-NAME= &(JV('#WORK1'))
/              ADD-FILE-LINK LINK=SORTOUT, -
/                                  FILE-NAME= &(OUTPUT-FILE-1)
/              SORT-FILE
/WORK-1-ERROR: IF-BLOCK-ERROR
/                  WRITE-TEXT 'Error on WORK-1 with SC1 = &(SC1)'
/                  HELP-MSG  &(MC)
/                  ELSE
/                  WRITE-TEXT 'WORK-1 terminated without error'
/                  END-IF
/WORK-2:      "Sorting file 2"
/              WAIT-EVENT  JV(COND = ( (&(MONJV-2),1,2) = '$T' ),-
/                                  TIME-LIMIT = 3600 )
/              CRE-JV      JV = #WORK2
/              MOD-JV      JV = #WORK2 , -
/                                  SET-VALUE = ( &(MONJV-2),129,54 )
/              ADD-FILE-LINK LINK=SORTIN, -
/                                  FILE-NAME= &(JV('#WORK2'))
/              ADD-FILE-LINK LINK=SORTOUT, -
/                                  FILE-NAME= &(OUTPUT-FILE-2)
/              SORT-FILE
/WORK-2-ERROR: IF-BLOCK-ERROR
/                  WRITE-TEXT 'Error on WORK-2 with SC1 = &(SC1)'
/                  HELP-MSG  &(MC)
/                  ELSE
/                  WRITE-TEXT 'WORK-2 terminated without error'
/                  END-IF

```


CANCEL-PRINT-JOB

Cancel print job

Description status:	SPOOL V4.9A
Functional area:	Controlling spoolout job
Domain:	SPOOL-PRINT-SERVICES SPOOL-PRINT-ADMINISTRATION
Privileges:	STD-PROCESSING PRINT-SERVICE-ADMINISTRATION OPERATING SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

The CANCEL-PRINT-JOB command cancels print jobs on any server in any cluster. Depending on the user group, print jobs can be canceled in local clusters or in both local and remote clusters.

Privileged functions

RSO device administrators and spool and cluster administrators can cancel any job scheduled for printing on any device that they manage. For more information on these user groups see also the manuals “RSO” [32], “SPOOL” [43] and “Distributed Print Service” [10].

9025/9026-RENO printers

After a PRINT-DOCUMENT command, the print job can be in one of three possible statuses. The CANCEL-PRINT-JOB command affects the print job in different ways, depending on the status of the print job:

Status of the print job	Effects of the CANCEL-PRINT-JOB command
Queued	Print job is deleted and nothing is printed
File transfer from BS2000 to 9025/9026-RENO memory started but not yet completed	Spoolin is interrupted: that part of the print file which has already been transferred or has passed through spoolin is printed; the print job is canceled
File transfer from BS2000 to 9025/9026-RENO memory completed, spoolout started or already completed	Command has no effect; no error message; the print job is executed in full

Format

CANCEL-PRINT-JOB
<p>JOB-IDENTIFICATION = *TSN (...) / *SERVER-TSN(...) / *MONJV(...) / *FOREIGN(...) / *SYSTEM-FILE(...)</p> <p>*TSN(...)</p> <ul style="list-style-type: none"> TSN = <alphanum-name 1..4> , CLUSTER-NAME = *LOCAL-CLUSTER / <alphanum-name 1..8> <p>*SERVER-TSN(...)</p> <ul style="list-style-type: none"> TSN = <alphanum-name 1..4> , SERVER-NAME = <alphanum-name 1..8> <p>*MONJV(...)</p> <ul style="list-style-type: none"> MONJV = <filename 1..54 without-gen-vers> <p>*FOREIGN(...)</p> <ul style="list-style-type: none"> IDENTIFICATION = <c-string 1..255 with-low> , CLUSTER-NAME = <alphanum-name 1..8> <p>*SYSTEM-FILE(...)</p> <ul style="list-style-type: none"> NAME = list-poss(16): *SYSOUT / *SYSLST(...) *SYSLST(...) SYSLST-NUMBER = *STD / <integer 1..99>

Operands

JOB-IDENTIFICATION = ***TSN**(...) / ***SERVER-TSN**(...) / ***MONJV**(...) / ***FOREIGN**(...) / ***SYSTEM-FILE**(...)

Type of job identification.

JOB-IDENTIFICATION = ***TSN**(...)

The job is identified by its local task sequence number (TSN).

TSN = <alphanum-name 1..4>

TSN of the job to be canceled.

CLUSTER-NAME = ***LOCAL-CLUSTER** / <alphanum-name 1..8>

Name of the cluster in which the print job is being processed. If a cluster name is specified, the specified TSN is the TSN on the gateway host in the specified remote cluster. Only a BS2000 cluster can be specified.

JOB-IDENTIFICATION = *SERVER-TSN(...)

The print job is identified by its TSN on the server. Only print jobs in the local cluster can be addressed in this way.

TSN = <alphanum-name 1..4>

TSN of the job to be canceled on the server.

SERVER-NAME = <alphanum-name 1..8>

Name of the server on which the print job can be addressed by means of its TSN.

JOB-IDENTIFICATION = *MONJV(...)

The print job is identified by its MONJV.

MONJV = <filename 1..54 without-gen-vers>

This operand can be used to address a print job by means of its MONJV, provided the specified MONJV is accessible on the host at which the command is issued. Only print jobs in the local cluster can be addressed in this way.

JOB-IDENTIFICATION = *FOREIGN(...)

The print job is identified by a non-BS2000 job name.

IDENTIFICATION = <c-string 1..255 with-lower-case>

This operand can be used to address print jobs being processed in a cluster of UNIX-based systems.

CLUSTER-NAME = <alphanum-name 1..8>

Name of the cluster in which the print job is being processed. See the "Distributed Print Service" manual [10].

JOB-IDENTIFICATION = *SYSTEM-FILE(...)

Identifies the system files *SYSOUT and *SYSLST. This operand resets an earlier PRINT-DOCUMENT ...,START-PROCESSING=*AT-FILE-CLOSING/integer command. The system file is then printed with the default attributes, not those specified for in the PRINT-DOCUMENT command.

NAME = list-poss(16): *SYSOUT / *SYSLST(...)

Specifies the system files *SYSOUT and/or *SYSLST.

NAME = *SYSLST(...)

Specifies the *SYSLST system file.

SYSLST-NUMBER = *STD / <integer 1..99>

The default SYSLST file or one or more numbers of the desired SYSLST files can be specified.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	No errors Guaranteed messages: SCP0891, SCP1031, SCP1032
2	0	SCP1034	No response from operator
2	0	SPS0178	File lock cannot be canceled Guaranteed message: SPS0178
2	0	SPS0455	MONJV processing error Guaranteed message: SPS0455
2	0	SPS0464	JV subsystem not loaded
2	0	SPS0469	MONJV validity problem Guaranteed message: SPS0469
2	0	SCP0892	TSN not found or command processing not permitted
4	32	SCP0974	Memory error or system error. Command rejected
4	64	SCP0976	Invalid operand value
6	128	CMD2241	DPRINTCL subsystem not loaded
1	128	SPS0266	SPOOL subsystem not loaded/SPOOL task not available

Notes

- Users must use this command to cancel print jobs they have sent to a different cluster (since local copies are not kept of print jobs sent to a different cluster). For example, the command /CANCEL-PRINT-JOB (TSN=12AB,CLUSTER-NAME=C1) cancels job 12AB in cluster C1.
- For their own print jobs in their own cluster, users can also use the existing command CANCEL-JOB, since there will be at least one local copy of such print jobs.
- The cluster administrator can cancel any print job in his or her own cluster. For example, to cancel a print job with a TSN of 1234 on remote server S1 that was issued on a remote client (i.e. when there is no local copy on the host at which the cluster administrator issues the CANCEL-PRINT-JOB command), the cluster administrator issues the command CANCEL-PRINT-JOB *SERVER-TSN(1234,S1).
- A server need not be specified to cancel a print job in a remote cluster. In the remote cluster, copies of all the print jobs that come back from the local cluster are kept on a selected server that receives all print jobs from remote clusters.
- The command must be used if a print job needs to be canceled on a remote cluster, because no copy of the print job is kept locally.

CANCEL-PROCEDURE

Cancel procedure run

Description status:	SYSFILE V19.0A
Functional area:	Procedures
Domain:	PROCEDURE
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

The CANCEL-PROCEDURE command cancels a procedure run. SYSCMD is assigned to the primary command input. (See the EXIT-PROCEDURE command: SYSCMD is assigned to the procedure most recently left). All system files opened during procedure execution, including TASKLIB, are closed and again receive their primary assignments.

Format

CANCEL-PROCEDURE	Alias: CNP

The CANCEL-PROCEDURE command has no operands and is executed immediately.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
	32	SSM1013	Command not executed due to system error
	64	SSM2039	Error on closing output system file; the SYSOUT message contains the DMS error codes as an insert

Notes

- Commands in procedure files: if an error in a non-S procedure triggers the spin-off mechanism, processing branches to the next of the following commands: CANCEL-PROCEDURE, LOGOFF, EXIT-JOB, SET-JOB-STEP, END-PROCEDURE or EXIT-PROCEDURE.
- A CANCEL-PROCEDURE command at procedure level 1 (i.e. no nesting) is identical to the EXIT-PROCEDURE command (see [figure 2](#)).

Examples

Example 1

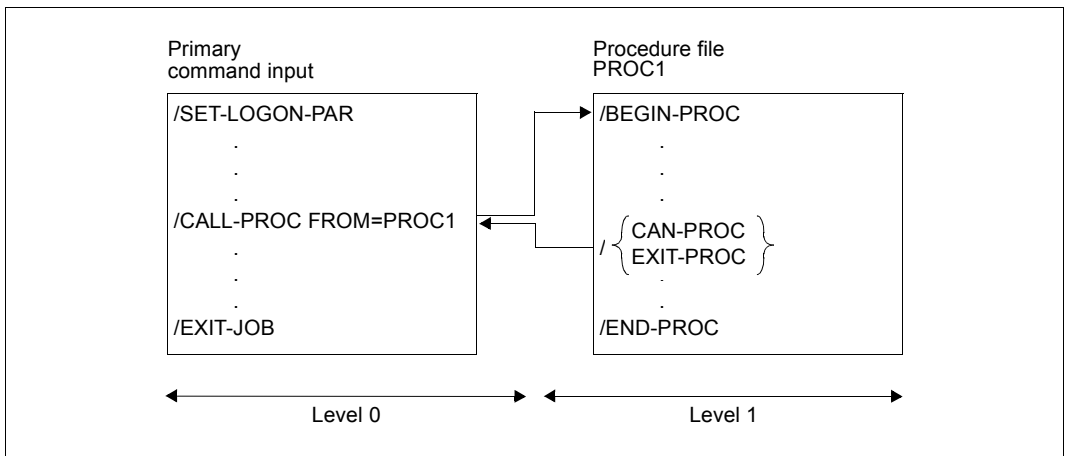


Figure 2: CANCEL- and EXIT-PROCEDURE commands at procedure level 1

The arrows indicate the sequence in which the commands are processed.

Example 2

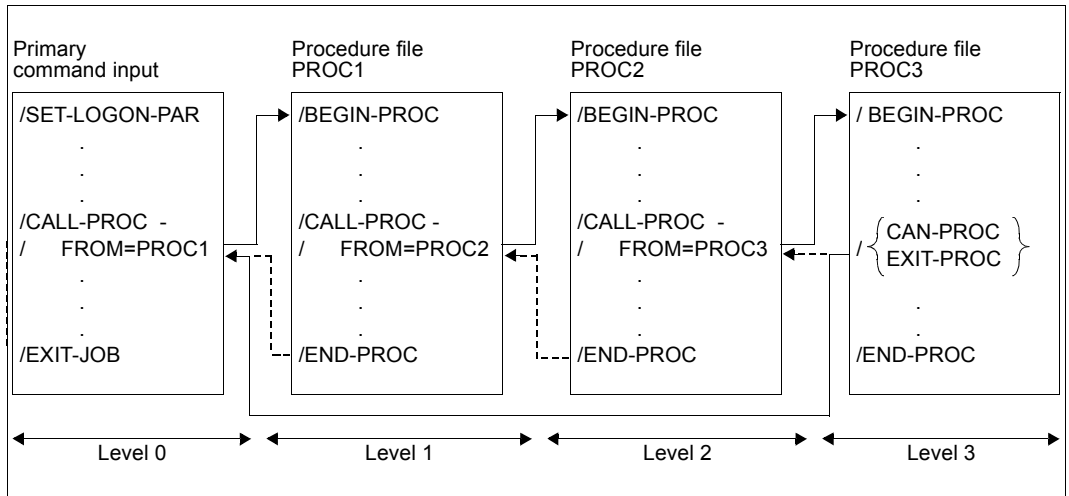


Figure 3: CANCEL- and EXIT-PROCEDURE commands at procedure level 3

The solid lines indicate the sequence in which the commands are processed in the case of CANCEL-PROCEDURE. By way of comparison, the broken lines indicate the process in the case of EXIT-PROCEDURE.

CANCEL-PROGRAM

Cancel program run

Description status:	BLSSERV V2.8A
Functional area:	Program control
Domain:	PROGRAM
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

The CANCEL-PROGRAM command cancels a program run and releases all the user memory reserved until that time by the dynamic binder loader DBL and the static loader ELDE.

If the program contains an STXIT routine for the ABEND event class, it is activated. The status indicator of any program-monitoring job variable is set to '\$A'.

Format

CANCEL-PROGRAM	Alias: CNPG

The CANCEL-PROGRAM command has no operands and is executed immediately.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
2	0	BLS0554	Warning: no program was loaded
	64	CMD0216	User has no authorization for the command
	64	BLS0553	The loaded program may not be unloaded

CANCEL-PUBSET-EXPORT

Cancel pubset export

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING
Routing code:	R

Function

This command cancels the wait state of an active EXPORT job that is to control the export of a pubset. The job terminates with an error message and the pubset remains available.

Format

CANCEL-PUBSET-EXPORT

PUBSET = <cat-id 1..4>

Operands

PUBSET = <cat-id 1..4>

Catalog ID of the pubset to be exported.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
1	0	DMS0355	Same export already active
1	0	DMS0364	Pubset not available
1	0	DMS036C	No task for EXPORT-PUBSET
	1	CMD0202	Syntax error
	32	DMS0363	Error while accessing MRSCAT
	64	DMS0360	No authorization for command
	130	DMS0351	Other import/export task active

CANCEL-PUBSET-IMPORT

Cancel import of pubset

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING
Routing code:	R

Function

The CANCEL-PUBSET-IMPORT command causes an active import job to be canceled. In particular in a shared pubset network, this function can be used to cancel an import where the master is being changed so that the pubset becomes accessible again from at least one other system.

When this command is called, the import job loses its current disk accessing rights and it is prevented from building up new disk accessing rights.

However, the import job can only terminate if it encounters an error due to the loss of rights or if it finds the cancel request that is stored in the MRS catalog. If the import job does not terminate, it is not possible to import the pubset in the current session. The data consistency of the pubset is, however, ensured when it is imported on another system.

Format

CANCEL-PUBSET-IMPORT

PUBSET = <cat-id 1..4>

Operands

PUBSET = <cat-id 1..4>

Catalog ID of the pubset to be imported.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD0202	Syntax error
	32	DMS13D1	CANCEL-PUBSET-IMPORT was canceled because an error occurred when accessing the volume catalog.
	64	DMS13D1	Error during CANCEL-PUBSET-IMPORT; the insert describes the cause of the error, e.g.: <ul style="list-style-type: none">– MRSCAT does not exist– no pubset import– a CANCEL-PUBSET-IMPORT is already being processed– the pubset has already been imported– the pubset import has already reached its final stage

CANCEL-RUN-PROCESS

Cancel processing of command file

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Editing command files
Domain:	not allocated
Privileges:	OPERATING
Routing code:	E

Function

The CANCEL-RUN-PROCESS command cancels processing of a command file that has already been started. In order to execute it, the operator must specify a RUN-ID. This is assigned to each command file started by means of the RUN command. Command files that are started from another RUN sequence are assigned the same RUN-ID as the generating command file. When a command file is canceled by means of CANCEL-RUN-PROCESS all the command files generated by it are canceled automatically.

This command must not necessarily be entered at the same console as the RUN command. However, it does require the routing code E.

The command must not be issued from a user task with the operating privilege.

Restrictions when using the "Operator LOGON" function

If the "Operator LOGON" function is used (system parameter NBCONOPI=Y), the command is not accepted after "SYSTEM READY" unless the originating task is running under TSOS or under the user ID which entered the RUN command. The command is also allowed if the RUN command was issued by an authorized user program with generated authorization names or if the RUN command issuer has been disconnected. In all other cases, if this condition is not met, the command is rejected with message NBR0002.

Format

CANCEL-RUN-PROCESS

RUN-ID = <alphanum-name 1..4>

Operands

RUN-ID = <alphanum-name 1..4>

ID of the command file to be canceled. A unique RUN-ID is assigned to each command file started by means of the RUN command. If no RUN-ID or an unknown RUN-ID is specified, the command is rejected.

For information on using command files, see the section on “Command files for the operator” in the “Introduction to System Administration” [14].

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD0202	Syntax error
	64	NBR0001	Specified RUN-ID not found
	64	NBR0002	No authorization

CHANGE-ACCOUNTING-FILE

Change system accounting file

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Accounting system control
Domain:	ACCOUNTING
Privileges:	TSOS

Function

The current accounting file, in which all accounting-specific data is stored, is closed and a new file is opened. After the accounting file has been changed, the old accounting file can still be accessed during the session for evaluation purposes, e.g. with the product RAV.

The default value *UNCHANGED in the corresponding operand means that the current specification remains valid.

Format

CHANGE-ACCOUNTING-FILE

```

NAME = *NEXT / *STD / <filename 1..54> / <partial-filename 2..53>
, SPACE = *UNCHANGED / *STD / *RELATIVE(...)
  *RELATIVE(...)
    | PRIMARY-ALLOCATION = <integer 1..65535>
    | , SECONDARY-ALLOCATION = <integer 0..32767>
, BUFFER-LENGTH = *UNCHANGED / *BY-PROGRAM / *STD(...)
  *STD(...)
    | SIZE = 1 / <integer 1..16>
, VOLUME = *UNCHANGED / *STD / <vsn 1..6>

```

Operands

NAME =

Defines the name of the accounting file to be opened.

NAME = *NEXT

The name of the continuation file is taken over for the new accounting file. The continuation file is determined:

- from the START-ACCOUNTING command if a list of alternative file names was defined in the ALTERNATE-FILES operand,
- by automatic generation of the file name by incrementing the current number by 1 (prerequisite: the old accounting file name was formed automatically).

NAME = *STD

The new accounting file receives the standard file name
\$TSOS.SYS.ACCOUNT.<date>.xxx.nn.

This file name is made up of the following elements:

<date> : yyyy-mm-dd date (year number as four digits)
 or
 yy.mm.dd date (year number as two digits)

xxx : session number

nn : current number of the accounting file

NAME = <filename 1..54>

Specifies a fully qualified name for the accounting file to be opened.

NAME = <partial-filename 2..53>

Specifies a partially qualified file name.

This specification results in automatic generation of the file name. The following points must be observed when specifying a partially qualified file name:

- the partial qualification must not exceed 26 characters in length (not including the user ID) because <date>.xxx.nn is added to the name; for catalog IDs that consist of more than one character, this value is reduced by the number of extra characters; if four-digit year numbers are being used (class 2 system parameter FMTYFNLG), only 24 characters may be used.
- the partial qualification may be terminated with the user ID; the file name automatically receives the suffix SYS.ACCOUNT.<date>.xxx.nn
- if no user ID is specified, the file is cataloged under TSOS.

SPACE =

Defines the storage space allocation for the file on the disk.

SPACE = *STD

The storage space for the file is 48 PAM blocks each for primary and secondary allocation.

SPACE = *RELATIVE(...)

Relative storage space allocation for the file.

PRIMARY-ALLOCATION = <integer 1..65535>

Number of PAM blocks for primary allocation.

SECONDARY-ALLOCATION = <integer 0..32767>

Number of PAM blocks for subsequent file extensions.

BUFFER-LENGTH = *UNCHANGED / *BY-PROGRAM / *STD(...)

Defines the block size for the input/output buffer of the accounting file.

Specification of this operand is meaningful only for new files and is otherwise ignored.

BUFFER-LENGTH = *BY-PROGRAM

Specifies a buffer of 2048 bytes (1 PAM block) for input/output of the file.

BUFFER-LENGTH = *STD(...)**SIZE = 1 / <integer 1..16>**

The specified number of PAM blocks is to be used as the buffer size. If the accounting file is to be set up on an NK4 pubset, the value specified should be an even number.

However, any odd number will automatically be converted to the next highest multiple of 2 when the command is processed. In such cases, a message will inform the user of this action.

VOLUME =

Specifies that the new accounting file is to be created on a certain volume.

VOLUME = *STD

The Data Management System of BS2000 decides on which volume the accounting file will be created.

VOLUME = <vsn 1..6>

The accounting file is to be created on the volume with the specified volume serial number. The VOLUME operand supports only volumes that do not require device-specific information.

If the accounting file is to be created on a **tape** or on a **private disk**, the system must be informed with a CREATE-FILE command before the accounting file is changed.

If in the case of a **public disk** this does not belong to the default pubset of the designated user ID, the catalog ID of the volume must be included in the file name. If the complete name is to be generated automatically, :catid:\$TSOS. must be specified.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
2	0	NAM3001	Requested action has been performed, but was accompanied by a warning
	1	CMD0202	Syntax error
	32	CMD0221	Internal system error
	64	NAM0012	No authorization for command
	64	NAM3003	Semantic error
	128	CMD2280	Temporarily unable to execute the command

CHANGE-CONSLOG-FILE

Close current logging file and open new file

Description status: BS2000 OSD/BC V10.0A
Functional area: Error logging
Domain: ERROR-LOGGING
SECURITY-ADMINISTRATION
Privileges: OPERATING
SAT-FILE-MANAGEMENT
TSOS

Function

The CHANGE-CONSLOG-FILE command closes the current logging file and opens a new one (see also “Logging file” in the “Introduction to System Administration” [14]).

This enables the closed logging files to be evaluated during the current session. Read access to the logging file currently open is obtained with the SET-CONSLOG-READ-MARK command.

If a CHANGE-CONSLOG-FILE command is already being processed, all further CHANGE-CONSLOG-FILE commands are rejected. Depending on the NBKESNR system parameter, up to 99 files can be created per session or 999 per day. The NBKESNR system parameter also governs whether the CONSLOG file is created under the TSOS or SYSAUDIT user ID. The NBLOGENF system parameter (NBLOGENF=E(nforced)) can be used to prevent the last possible CONSLOG file from being changed.

When printing the closed logging file it is advisable to specify the LAST-CHARACTER=252 operand in the PRINT-DOCUMENT command.

Format

CHANGE-CONSLOG-FILE

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
1	0	NBR0905	CONSLOG not active
2	0	NBR0906	CONSLOG being deactivated
	64	NBR0904	Last possible sequence number reached, command no longer possible
	130	EXC065A	CHANGE-CONSLOG already being processed

CHANGE-DISK-MOUNT

Lock private disk

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Device management
Domain:	DEVICE
Privileges:	OPERATING
Routing code:	D

Function

This command denies the user access to a private disk already in use.

Format

CHANGE-DISK-MOUNT

```

UNIT = *VOLUME(...) / *EXCHANGE-PAIR(...) /
      list-poss(10): <alphanum-name 2..2> / <alphanum-name 4..4>
*VOLUME(...)
  | VOLUME = list-poss(10): <vsn 1..6>
*EXCHANGE-PAIR(...)
  | EXCHANGE-PAIR = list-poss(10): *PARAMETER(...)
    | *PARAMETER(...)
      | DISMOUNT-VOLUME = <vsn 1..6>
      | REMOUNT-VOLUME = <vsn 1..6>
,ACTION = *EXCHANGE / *MOVE / *CANCEL

```

Operands

UNIT =

Specifies one or more disks which are to be no longer available.

UNIT = *VOLUME(...)

Specifies the volume serial numbers (VSN, up to 6 characters) of one or more disks which are to be no longer available. A maximum of 10 disks may be specified.

VOLUME = list-poss(10): <vsn 1..6>

Specifies the volume serial number (VSN).

UNIT = *EXCHANGE-PAIR(...)

This operand value is not relevant to BS2000/OSD-BC V1.0 and above, as in this and subsequent versions on only hard disk devices are supported.

UNIT = list-poss(10): <alphanum-name 2..2> / <alphanum-name 4..4>

Lists up to 10 disks that are to be no longer available.

ACTION = *EXCHANGE / *MOVE / *CANCEL

Specifies the way in which the mount state is to be changed.



The operand values *EXCHANGE and *MOVE are not relevant to BS2000/OSD-BC V1.0 and above, as in this and subsequent versions only hard disk devices are supported.

ACTION = *CANCEL

The allocated private disk specified under UNIT in the *VOLUME operand is no longer to be available to the user.

Every I/O request is rejected. Opened files are not closed; these files must be made available again using the REMOVE-FILE-ALLOCATION-LOCKS command before the next OPEN.

The disk parameters ASSIGN-TIME=*USER and USER-ALLOCATION=*NO are implicitly set. In this way all allocation requests are rejected until the disk is released by the occupying users. The disk can only be made available again (SET-DISK-VOL=...,USER=*ALL) after it has been released by all users. Until that time it is also defined as allocated in the SHOW commands (SH-DEV, SH-DISK).



The CANCEL function is an “emergency command” and should thus be used only in urgent cases, such as with a permanent INOP for the disk.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	NKV0001	Syntax error
	64	NKV0004	Command partially processed
	64	NKV0005	Command not processed for an object
	64	NKV0006	Command not processed
	130	NKVD002	Disk monitor not available

CHANGE-FILE-LINK

Change file link name

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT

Function

The CHANGE-FILE-LINK command changes the file link name in an existing entry in the Task File Table (TFT), i.e. it assigns a new file link name to the file. All the other values in this TFT entry will remain unchanged.

The CHANGE-FILE-LINK command cannot be applied to the TFT entry of a file that is currently open. The ADD-FILE-LINK command can be used to create a new TFT entry; the SHOW-FILE-LINK command returns information on existing TFT entries.

Format

CHANGE-FILE-LINK	Alias: CGFL
LINK-NAME = * FIRST-BLANK / <filename 1..8 without-gen> , NEW-NAME = <filename 1..8 without-gen>	

Operands

LINK-NAME = *FIRST-BLANK / <filename 1..8 without-gen>

Specifies the existing link name of the file.

LINK-NAME = *FIRST-BLANK

If *FIRST-BLANK is specified, the first TFT entry with a link name consisting of blanks is selected for change. Such an entry is created if neither the link name nor the file name is defined in the FCB macro call of a program and if no appropriate ADD-FILE-LINK command is issued before the program call. The file name in this entry is the symbolic name of the FCB macro call.

LINK-NAME = <filename 1..8 without-gen>

Specifies the existing link name of the file.

NEW-NAME = <filename 1..8 without-gen>

The new link name to be given to the file.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
	1	CMD0202	Syntactical or semantic error in command
	64	DMS05D5	Specified LINK name missing or invalid
	64	DMS05DD	New LINK name already exists
	64	DMS06FF	BCAM connection severed
	130	DMS0582	File is currently locked or being used and cannot be processed

Example*Changing a file link name*

```

/show-file-attr max.file.2,inf=(org=*yes) _____ (1)
%0000000003 :20SG:$USER1.MAX.FILE.2
% ----- ORGANIZATION -----
% FILE-STRUC = NONE          BUF-LEN = NONE          BLK-CONTR = NONE
% IO(USAGE)  = READ-WRITE   IO(PERF) = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = NONE          REC-SIZE = 0
% AVAIL      = *STD
% WORK-FILE  = *NO          F-PREFORM = *K          SO-MIGR   = *ALLOWED
%:20SG: PUBLIC:      1 FILE RES=      3 FRE=      3 REL=      3 PAGES

/add-file-link link=pcin,file-name=max.file.2 _____ (2)

/add-file-link link=pcout,file-name=max.file.9,access-method=*isam,
support=*disk(isam-attr=(key-pos=10,key-len=4)) _____ (3)

/start-percon _____ (4)
% BLS0523 ELEMENT 'PCROOT', VERSION '029', TYPE 'L' FROM LIBRARY ':10SH:$TSOS.S
YSLNK.PERCON.029' IN PROCESS
% BLS0524 LLM 'PCROOT', VERSION '02.9A' OF '2008-10-06 14:27:00' LOADED
% BLS0551 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2009. ALL RIGHTS RESERVED
% PER0000 PERCON STARTED, VERSION V02.9A10
%//assign-input-file file=*disk-file _____ (5)

%//assign-output-file file=*disk-file _____ (6)

%//start-conv _____ (7)
% PER0030 NUMBER OF PROCESSED RECORDS FOR LINK='PCIN' (FILE=:20SG:$USER1.MAX
.FILE.2): 9
% PER0030 NUMBER OF PROCESSED RECORDS FOR LINK='PCOUT' (FILE=:20SG:$USER1.MA
X.FILE.9): 9
%//end _____ (8)
% PER0031 PERCON TERMINATED NORMALLY

/show-file-link _____ (9)
% PCIN :20SG:$USER1.MAX.FILE.2
% PCOUT :20SG:$USER1.MAX.FILE.9
/show-file-link link=pcout,inf=(file-control-block=*yes) _____ (10)
%
%-- LINK-NAME ----- FILE-NAME -----
% PCOUT :20SG:$USER1.MAX.FILE.9

```

```

% ----- FILE-CONTROL-BLOCK - GENERAL ATTRIBUTES -----
% ACC-METH = ISAM          OPEN-MODE = *BY-PROG  REC-FORM = *BY-PROG
% REC-SIZE = *BY-PROG     BUF-LEN   = *BY-PROG  BLK-CONTR = *BY-PROG
% F-CL-MSG = STD         CLOSE-MODE = *BY-PROG
% ----- FILE-CONTROL-BLOCK - DISK FILE ATTRIBUTES -----
% SHARED-UPD = *BY-PROG  WR-CHECK = *BY-PROG  IO(PERF) = *BY-PROG
% IO(USAGE)  = *BY-PROG  LOCK-ENV  = *BY-PROG
% ----- FILE-CONTROL-BLOCK - TAPE FILE ATTRIBUTES -----
% LABEL      = *BY-PROG  (DIN-R-NUM = *BY-PROG, TAPE-MARK = *BY-PROG)
% CODE       = *BY-PROG  EBCDIC-TR  = *BY-PROG  F-SEQ      = *BY-PROG
% CP-AT-BLIM = *BY-PROG  CP-AT-FEOV = *BY-PROG  BLOCK-LIM = *BY-PROG
% REST-USAGE = *BY-PROG  BLOCK-OFF  = *BY-PROG  TAPE-WRITE = *BY-PROG
% STREAM     = *BY-PROG
% ----- FILE-CONTROL-BLOCK - ISAM FILE ATTRIBUTES -----
% KEY-POS    = 10        KEY-LEN    = 4          POOL-LINK  = *BY-PROG
% LOGIC-FLAG = *BY-PROG  VAL-FLAG  = *BY-PROG  PROPA-VAL  = *BY-PROG
% DUP-KEY    = *BY-PROG  PAD-FACT  = *BY-PROG  READ-I-ADV = *BY-PROG
% WR-IMMED   = *BY-PROG  POOL-SIZE  = *BY-PROG

/show-file-attr max.file.9,inf=(org=*yes) _____ (11)
%00000003 :20SG:$USER1.MAX.FILE.9
% ----- ORGANIZATION -----
% FILE-STRUC = ISAM      BUF-LEN   = STD(1)      BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE IO(PERF) = STD        DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)     REC-SIZE  = 0
% KEY-LEN    = 4         KEY-POS   = 10
% AVAIL      = *STD
%:20SG: PUBLIC:      1 FILE RES=      3 FRE=      1 REL=      0 PAGES

/change-file-link link=pcout,new-name=editisam _____ (12)

/show-file-link _____ (13)
%-- LINK-NAME ----- FILE-NAME -----
% EDTISAM           :20SG:$USER1.MAX.FILE.9
% PCIN              :20SG:$USER1.MAX.FILE.2

/start-edt _____ (14)
.
.
.   "Processing with the EDT utility routine"
.
.

/remove-file-link link=editisam _____ (15)

```

- (1) Output of the organization attributes of the SAM file *MAX.FILE.2*.
- (2) Creates a TFT entry for the file *MAX.FILE.2* under the link name *PCIN* (default link name of the utility routine PERCON for the input file).
- (3) Creates a TFT entry for the file *MAX.FILE.9* under the link name *PCOUT* (default link name of the utility routine PERCON for the output file). The file *MAX.FILE.2* is to be converted with the PERCON utility to the ISAM file *MAX.FILE.9* by constructing the ISAM key from the data bytes 6 to 9.
- (4) Invocation of the PERCON utility routine.
- (5) The file that is entered in the TFT with the default link name *PCIN* is to be used as the input file.

- (6) The file that is entered in the TFT with the default link name *PCOUT* is to be used as the output file.
- (7) The conversion begins.
- (8) The PERCON utility routine is terminated.
- (9) The output of the TFT entries indicates that the entries with link names *PCIN* and *PCOUT* still exist.
- (10) Output of the TFT entry with the link name *PCOUT* showing the file organization attributes.
- (11) Output of the file organization attributes from the catalog entry of the file *MAX.FILE.9*.
- (12) In order to enable processing of the file *MAX.FILE.9* with the utility routine EDT, the link name *PCOUT* is changed to *EDTISAM* with the CHANGE-FILE-LINK command. This makes it possible to process the existing TFT entry for the file *MAX.FILE.9* with the utility routine EDT via the default link name *EDTISAM* (without the TFT entry, EDT only processes ISAM files with KEY-POS=5 and KEY-LENGTH=8). It is not necessary to create a new TFT entry with ADD-FILE-LINK.
- (13) The output of the TFT entries indicates that the entries with the link names *PCIN* and *EDTISAM* exist.
- (14) Invocation of the utility routine EDT.
- (15) After processing in the EDT is complete, the TFT entry with the link name *EDTISAM* is deleted.

CHANGE-HEL-FILE

Change hardware error logging file

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Error logging
Domain:	ERROR-LOGGING
Privileges:	HARDWARE-MAINTENANCE

Function

The CHANGE-HEL-FILE closes the current HEL (Hardware Error Logging) file and opens a new logging file.

Hardware error logging is initiated either automatically on system startup or explicitly with the START-HEL-LOGGING command. The CHANGE-HEL-FILE command is rejected if logging is not active (message HEL0010).

HEL logging files are created under the TSOS user ID with a default name formed on the pattern \$TSOS.SYS.HEL.*yyyy-mm-dd.hhmmss*. The name components *yyyy-mm-dd* and *hhmmss* respectively identify the date and time when the logging file was initially opened. The file access rights are by default set to USER-ACCESS=*SPECIAL and ACCESS=*READ, which means that, in addition to TSOS, all user IDs including the maintenance IDs (HARDWARE-MAINTENANCE privilege) have read access to HEL logging files. A HEL logging file is given the size specified in the SPACE operand. There is no provision for dynamic extension (SECONDARY-ALLOCATION=0). If the file grows to the size of the primary allocation, it is automatically closed and a new logging file is created and opened.

Format

CHANGE-HEL-FILE
SPACE = <u>*UNCHANGED</u> / *STD / <integer 24..65536 2Kbyte>

Operands

SPACE = *UNCHANGED / *STD / <integer 24..65536>

Defines the size (primary allocation, plus rounding as required; see the PRIMARY-ALLOCATION operand of the CREATE-FILE command) of the logging file that is to be created.

SPACE = *UNCHANGED

The logging file is to be the same size as the old one.

SPACE = *STD

The logging file is given the default size of 400 PAM pages.

SPACE = <integer 24..65536 2Kbyte>

The logging file is given the specified number of PAM pages.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	32	HEL0100	Internal error
	32	HEL0105	Bourse problems
	32	HEL0106	DMS error
	32	HEL0107	DMS error
	64	HEL0010	Hardware error logging nor active
	64	HEL0110	Privilege error

Example

```
/change-hel-file
% HEL0001 HW-ERROR-LOGGING FILE ':1SBZ:$TSOS.SYS.HEL.2012-01-29.141847'
OPENED WITH 'SPACE= 402'
```

The CHANGE-HEL-FILE command with no operands opens a new logging file of the default size (in this case rounded up to 402 PAM pages; see the SPACE operand).

CHANGE-RESLOG-FILE

Change RESLOG file

Description status:	RESLOG V1.7A
Functional area:	Accounting system control
Domain:	ACCOUNTING
Privileges:	TSOS SW-MONITOR-ADMINISTRATION

Function

The RESLOG (RESource LOGging) subsystem always logs the use of extra CPUs in the RESLOG file \$TSOS.SYS.RESLOG.<server-id>. You can use the CHANGE-RESLOG-FILE command to change the RESLOG file. Changing the logging file involves the following steps:

1. Close the current logging file \$TSOS.SYS.RESLOG.<server-id>
2. Rename the file to \$TSOS.SYS.RESLOG.<server-id>.<date>
(the close time date is appended to the file name)
3. Create and open the new logging file \$TSOS.SYS.RESLOG.<server-id>

Closed RESLOG files can be evaluated using the START-RESLOG-EVALUATION command.

RESLOG logging is only performed at systems with extra CPUs (e.g. systems of type S140 and S170). For more details on RESLOG logging and the use of extra CPUs see the “Introduction to System Administration” [14].

Format

CHANGE-RESLOG-FILE

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error



Command return codes that are returned in the event of an error (subcode 1 not equal to 0) cannot be guaranteed. Automatic error handling is nevertheless possible using the spin-off mechanism (corresponds to the default setting of the SET-PROCEDURE-OPTIONS command with ERROR-MECHANISM=*SPIN-OFF-COMPATIBLE).

CHANGE-SERSLOG-FILE

Change SERSLOG file

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Error logging
Domain:	ERROR-LOGGING
Privileges:	TSOS SW-MONITORING
Routing code:	A

Function

The CHANGE-SERSLOG-FILE command is executed only if software error logging is active.

The command closes the current SERSLOG file for logging relevant software errors (\$TSOS.SYS.SERSLOG.yyyy-mm-dd.xxx.nn) and opens a new one (with nn+1). The switchable logging procedure SERSLOG, which up to the time of command input has entered a record in the old file for each relevant software error, from now on will enter the records in the new file. The records automatically include the time, caller and further identifiers.

The information collected in the closed file can still be evaluated during the system session.

If the new SERSLOG file cannot be opened due to a DMS error, a message to that effect is issued at the operator terminal. In this case the old SERSLOG file remains the current file. If another CHANGE-SERSLOG-FILE command is issued, an attempt is made to open the **next-but-one** SERSLOG file. If the attempt succeeds, the number nn in the file name (see above) is incremented by 2 in order to mark the error in the sequence of files.

Format

CHANGE-SERSLOG-FILE

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
2	0	NER0000	Internal error
	32	CMD0221	System error
	32	NER0000	Internal error
	64	NER1070	Task break during processing
	64	NER1000	No authorization for command
	64	NER1040	DMS error
	128	NER0010	A SERSLOG command is already being processed
	129	NER1020	SERSLOG function is not active

Example

```

/show-serslog
% NER1060 SERSLOG ACTIVE. FILE : ':SBZ7:$TSOS.SYS.SERSLOG.2012-01-27.006.01'

/change-serslog-file

/show-serslog
% NER1060 SERSLOG ACTIVE. FILE : ':SBZ7:$TSOS.SYS.SERSLOG.2012-01-27.006.02'

```

CHANGE-STORAGE-CLASS-CATALOG

Change storage class catalog

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS

Functions

The CHANGE-STORAGE-CLASS-CATALOG command allows systems support personnel to change the storage class catalog.

The command can also be used to dynamically load the storage class catalog if it was corrupt when the SM pubset was imported and so could not be loaded at that time (message DMS1483).

The file named `:<sm-pubset-catid>:$TSOS.SYSCAT.STORCLS` contains the current storage class catalog. If the catalog is to be changed using CHANGE-STORAGE-CLASS-CATALOG, the new catalog must be stored as `SYSCAT.STORCLS.BAK` (with the same user and catalog IDs). When the command is executed, the original catalog is saved as `SYSCAT.STORCLS.<date>.<time>`, where `<date>` is the date (in yyyy-mm-dd format) and `<time>` is the time (in hhmmss format) when the catalog was changed. The new catalog is renamed to `SYSCAT.STORCLS` and thus becomes the current storage class catalog.

The procedure for dynamically loading and changing the storage class catalog is as follows:

1. Create a file named `SYSCAT.STORCLS.BAK` in the control volume set. It is best to create the file in VOLRES, so that the control volume set can also be reduced in size later.
2. Using HSMS, transfer the contents of a backup copy of the storage class catalog to `SYSCAT.STORCLS.BAK`.
3. Run the CHANGE-STORAGE-CLASS-CATALOG command. The storage class catalog will then be dynamically loaded from `SYSCAT.STORCLS.BAK`.

Format

CHANGE-STORAGE-CLASS-CATALOG

PUBSET = <cat-id 1..4>**Operands****PUBSET = <cat-id 1..4>**

Identifies the pubset for which the storage class catalog is to be changed or dynamically loaded.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
	1	CMD0202	Syntax error in command
	32	CMD0221	Internal system error
	64	CMD0216	No authorization for command
	64	DMS1485	Pubset not known
	64	DMS1486	Pubset is not a system-managed pubset
	64	DMS1487	Pubset not available
	64	DMS1490	Storage class management not available for this pubset
	64	DMS1483	Storage class catalog invalid
	64	DMS1481	Error accessing storage class catalog
	64	DMS1499	Catalog SYSCAT.STORCLS.BAK not in control volume set
	64	DMS1495	Catalog SYSCAT.STORCLS.BAK invalid
	64	DMS1496	Catalog SYSCAT.STORCLS.BAK does not exist
	129	DMS148D	Not enough class 4/5 memory
	129	DMS148E	Error on HIPLEX MSCF connection to master
	129	DMS148F	GCF subsystem not ready

CHANGE-TAPE-MOUNT

Change mount state of tape

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Device management
Domain:	DEVICE
Privileges:	OPERATING
Routing code:	T

Function

This command changes the mount state of a tape (initiates the remounting of a tape).

Format

CHANGE-TAPE-MOUNT

```

UNIT = *VOLUME(...) / *EXCHANGE-PAIR(...) /
      list-poss(10): <alphanum-name 2..2> / <alphanum-name 4..4>
*VOLUME(...)
  | VOLUME = list-poss(10): <vsn 1..6>
*EXCHANGE-PAIR(...)
  | EXCHANGE-PAIR = list-poss(10): *PARAMETER(...)
    | PARAMETER(...)
      | DISMOUNT-VOLUME = <vsn 1..6>
      | REMOUNT-VOLUME = <vsn 1..6>
,ACTION = *EXCHANGE / *MOVE / *CANCEL / *POSITION

```

Operands

UNIT =

Specifies one or more tapes to be remounted.

UNIT = *VOLUME(...)

Specifies the volume serial numbers (VSN, up to 6 characters) of one or more tapes whose mount state is to be changed. A maximum of 10 tapes may be specified.

VOLUME = list-poss(10): <vsn 1..6>

Specifies the VSN.

UNIT = *EXCHANGE-PAIR(...)

A used tape for which no device is available is to be exchanged for a used tape which is allocated a device. A maximum of 10 pairs may be specified.

EXCHANGE-PAIR = list-poss(10): *PARAMETER(...)

Specifies up to 10 pairs of used tapes. In each pair, one tape is to be exchanged for the other.

DISMOUNT-VOLUME = <vsn 1..6>

This used tape, which is allocated a device (PHASE IN-USE, ACTION neither NO DEVICE nor CANCELLED), is to be dismantled. Processing of this tape is interrupted but not terminated.

REMOUNT-VOLUME = <vsn 1..6>

This used tape, for which no device is available (PHASE='IN-USE' and ACTION≠'NO DEVICE' or ACTION≠'CANCELLED'), is to be mounted instead of the tape to which a device is allocated. Interruption processing of this tape is terminated. All input/output requests for this tape are processed again.

UNIT = list-poss(10): <alphanum-name 2..2> / <alphanum-name 4..4>

Lists up to 10 tapes to be remounted.

ACTION =

Specifies the way in which the mount state is to be changed.

ACTION = *EXCHANGE

In each of the tape pairs specified under UNIT in the *EXCHANGE-PAIR operand, one tape is to be exchanged for the other.

ACTION = *MOVE

The tape specified under UNIT in the *VOLUME operand or the tape on the device specified by means of its mnemonic name is to be remounted on another device. The system proposes a free device. For remounting the following requirements apply:

- there must exist one free device which supports the recording density used;
- the tape is in the PHASE='IN-USE' status and ACTION≠'NO DEVICE' or ACTION≠'CANCELLED',
- the user must not have specified NO MOVE at reservation time

ACTION = *CANCEL

The allocated tape specified under UNIT in the *VOLUME operand is no longer to be available to the user. All further input/output requests are automatically rejected until the tape has been released by the user.

ACTION = *POSITION

The allocated tape specified under UNIT in the *VOLUME operand is to be repositioned. In this way the operator can correct any inadvertent intervention on the device.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	NKV0001	Syntax error
	64	NKV0004	Command partially processed
	64	NKV0005	Command not processed for an object
	64	NKV0006	Command not processed
	130	NKVT002	Tape monitor not available

Notes

- Remounting (MOVE function) for a tape should always be announced with a command because if repositioning takes place without a command there is no guarantee that the tape on the backup device will remain repositionable.
- The state PHASE='IN-USE' and ACTION='NO DEVICE' can only occur for a tape if the device on which the tape is mounted is immediately detached and no backup device is available (detachment by the operator with /DETACH-DEVICE UNIT=mn , FORCE=YES or automatic detachment by the system).
- In the case of tapes without a STD label whose device is inoperable and for which no backup device is available, processing is stopped immediately.
The display ACTION='NO DEVICE' is not possible for these tapes.

CHANGE-TASK-CPU-LIMIT

Change maximum CPU time for task

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Job processing
Domain:	JOB
Privileges:	OPERATING TSOS
Routing code:	P

Function

The CHANGE-TASK-CPU-LIMIT command allows systems support to increase a task's maximum CPU time (task time limit, see also [section "Time limits in BS2000" on page 1-103](#)). This is a way of preventing important batch jobs being terminated prematurely because they have exceeded their CPU time limit.

Systems support can use the SHOW-JOB-STATUS command to check up on the elapsed time and the CPU time limit of a batch job which must not be terminated prematurely and can then run CHANGE-TASK-CPU-LIMIT to raise its CPU limit in good time.

When a batch job reaches its CPU time limit, message EXC0070 is displayed on the operator terminal. Systems support now has a maximum of 20 minutes in which to raise the job's CPU time limit using the CHANGE-TASK-CPU-LIMIT command. If the limit is not raised within this period, the batch job terminates.

If the CPU time limit is passed while a program is running, any STXIT routine defined for the program will not be processed unless the CPU limit is raised within 20 minutes.

Format

CHANGE-TASK-CPU-LIMIT
<p>JOB-IDENTIFICATION = *TSN(...) / *MONJV(...)</p> <p> *TSN(...) TSN = <alphanum-name 1..4></p> <p> *MONJV(...) MONJV = <filename 1..54 without-gen-vers></p> <p>,ADD = *SECONDS(...) / *PERCENT(...)</p> <p> *SECONDS(...) SECONDS = <integer 1..32767 seconds></p> <p> *PERCENT(...) PERCENT = <integer 1..100></p>

Operands

JOB-IDENTIFICATION =

Type of job identification.

Jobs may be identified by task sequence number (TSN) or by monitoring JV (MONJV).

JOB-IDENTIFICATION = *TSN(...)

The job is identified by its task sequence number (TSN).

TSN = <alphanum-name 1..4>

TSN of the job whose CPU time limit is to be raised.

JOB-IDENTIFICATION = *MONJV(...)

The job is identified by its monitoring JV.

MONJV = <filename 1..54 without-gen-vers>

Name of the JV monitoring the job whose CPU time limit is to be raised.

ADD = *SECONDS(...) / *PERCENT(...)

Specifies the amount by which the CPU time limit is to be raised. The increase can be given in seconds or as a percentage.

ADD = *SECONDS(...)

The increase is given in seconds.

SECONDS = <integer 1..32767 seconds>

Number of seconds by which the CPU time limit is to be raised.

ADD = *PERCENT(...)

The increase is given as a percentage of the current CPU time limit.

PERCENT = <integer 1..100>

Percentage by which the CPU time limit is to be raised.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	32	CMD0221	Internal error
	64	CMD0216	No authorization
	64	NTI6001	Parameter error; TSN or. MONJV unknown

CHANGE-TASK-PRIORITY

Change run priority of active job

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Job processing
Domain:	JOB
Privileges:	STD-PROCESSING OPERATING SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION
Routing code:	P

Function

The CHANGE-JOB-PRIORITY command changes the run priority of a job generated and running under the user's own ID. The command can also be used to process jobs generated by the user's own ID and running under a different user ID (extended access). Extended access is possible only in a homogeneous computer network. It is not possible in a nonhomogeneous computer network with BS2000 versions \leq BS2000/OSD-BC V1.0. The maximum permissible priority is the numeric minimum of the maximum values (i.e. the more favorable of the values) in the user catalog and the job class definition. If no maximum value is defined for the job class, then the standard run priority is used.

Information on the entry in the user catalog can be requested using the SHOW-USER-ATTRIBUTES command (output field MAX-RUN-PRIORITY), using the job class definition with the SHOW-JOB-CLASS (output field RUN PRIORITY) command.

The priority of jobs that have not yet been started can be changed using the MODIFY-JOB command.

Privileged function

If the CHANGE-TASK-PRIORITY command is issued at the operator terminal or under the TSOS user ID, it can be used to change the run priority of a job issued under any user ID. In this case the maximum permissible priority as defined for that user ID in the user catalog is irrelevant.

Format

CHANGE-TASK-PRIORITY

JOB-IDENTIFICATION = *TSN(...) / *MONJV(...)

*TSN(...)

TSN = <alphanum-name 1..4>

HOST = *STD / <c-string 1..8>

*MONJV(...)

MONJV = <filename 1..54 without-gen-vers>

,RUN-PRIORITY = <integer 30..255>

Operands

JOB-IDENTIFICATION =

Type of job identification.

Jobs may be identified by task sequence number (TSN) or by monitoring JV (MONJV).

JOB-IDENTIFICATION = *TSN(...)

The job is identified by its task sequence number (TSN) and where appropriate by the host on which it is running.

TSN = <alphanum-name 1..4>

TSN of the job whose priority is to be changed.

HOST = *STD / <c-string 1..8>

Host system on which the job is running.

The default is *STD, which means that the job is running on the local host.

The host name of a remote system can only be specified for systems which are on a computer network (see the "HIPLEX MSCF" manual [25]).

JOB-IDENTIFICATION = *MONJV(...)

The job is identified by its monitoring JV.

MONJV = <filename 1..54 without-gen-vers>

Name of the JV monitoring the job whose priority is to be changed.

RUN-PRIORITY = <integer 30..255>

Specifies the run priority to be assigned to the job. A low value indicates a high priority. The maximum permissible priority is the numeric minimum of the maximum values (i.e. the more favorable of the values) in the user catalog and the job class definition. If no maximum value is defined for the job class, the standard run priority is used. The values can be queried using the SHOW-USER-ATTRIBUTES and SHOW-JOB-CLASS commands.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD0202	Syntax error
	32	CMD0221	System error
	32	JMS0024	System error
	64	JMS0630	Semantic error (see SYSOUT message)
	64	JMS0640	MONJV error, illegal access or invalid time of call
	64	JMS0670	Error in a remote job
	130	JMS0620	Memory saturation
	130	JMS0650	MSCF or destination processor not available

CHANGE-VOLUME-SET-LIST-CATALOG

Change volume set list catalog

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS

Function

The CHANGE-VOLUME-SET-LIST-CATALOG command allows systems support to change the volume set list catalog.

The command can also be used to dynamically load the volume set list catalog if it was corrupt when the SM pubset was imported and so could not be loaded at that time (message DMS1484).

The file named `:<sm-pubset-catid>:$TSOS.SYSCAT.VSETLST` contains the current volume set list catalog. If the catalog is to be changed using CHANGE-VOLUME-SET-LIST-CATALOG, the new catalog must be stored as SYSCAT.VSETLST.BAK (with the same user and catalog IDs). When the command is executed, the original catalog is saved as SYSCAT.VSETLST.<date>.<time>, where <date> is the date (in yyyy-mm-dd format) and <time> is the time (in hhmmss format) when the catalog was changed. The new catalog is renamed to SYSCAT.VSETLST and thus becomes the current volume set list catalog.

The procedure for dynamically loading and changing the volume set list catalog is as follows:

1. Create a file named SYSCAT.VSETLST.BAK in the control volume set. It is best to create the file in VOLRES, so that the control volume set can also be reduced in size later.
2. Using HSMS, transfer the contents of a backup copy of the volume set list catalog to SYSCAT.VSETLST.BAK.
3. Run the CHANGE-VOLUME-SET-LIST-CATALOG command. The volume set list catalog will then be dynamically loaded from SYSCAT.VSETLST.BAK.

Format

CHANGE-VOLUME-SET-LIST-CATALOG

PUBSET = <cat-id 1..4>

Operands

PUBSET = <cat-id 1..4>

Designates the pubset for which the volume set list catalog is to be changed or dynamically loaded.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
	1	CMD0202	Syntax error in command
	32	CMD0221	Internal system error
	64	CMD0216	No authorization for command
	64	DMS1485	Pubset not known
	64	DMS1486	Pubset is not a system-managed pubset
	64	DMS1487	Pubset not available
	64	DMS1490	Storage class management not available for this pubset
	64	DMS1484	Volume set list catalog invalid
	64	DMS1482	Error accessing volume set list catalog
	64	DMS1499	Catalog SYSCAT.VSETLST.BAK not in control volume set
	64	DMS1495	Catalog SYSCAT.VSETLST.BAK invalid
	64	DMS1496	Catalog SYSCAT.VSETLST.BAK does not exist
	129	DMS148D	Not enough class 4/5 memory
	129	DMS148E	Error on HIPLEX MSCF connection to master
	129	DMS148F	GCF subsystem not ready

CHECK-DISK-MOUNT

Check mount state of disk

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Device management
Domain:	DEVICE
Privileges:	OPERATING
Routing code:	D

Function

This command checks the mount state of disk devices. The disk monitor (DM) is called to update the online state of the specified devices if the mounting or dismounting of a disk is not recognized by the system because of a missing activation interrupt.

Format

CHECK-DISK-MOUNT

UNIT = *ALL / list-poss(10): <alphanum-name 2..2> / <alphanum-name 4..4>
ACTION = *UPDATE / *REPORT

Operands

UNIT =

Defines the devices whose mount state is to be checked and updated.

UNIT = *ALL

All disk drives having the following attributes are checked:

- ATTACHED or DETACH-PENDING
- ACTION-STATE = NO_ACTION
- PHASE ≠ MOUNT and PHASE ≠ IN-USE.

UNIT = list-poss(10): <alphanum-name 2..2> / <alphanum-name 4..4>

Specifies the mnemonic device names (2 or 4 alphanumeric characters) of the devices whose mount state is to be checked and updated. A maximum of 10 devices may be specified.

ACTION =

Defines the logging of the online state.

ACTION = *UPDATE

An updated online state is to be logged by the following messages (default):

```
NKVD010 DISK vsn IS MOUNTED ON mn
NKVD011 DISK vsn IS DISMOUNTED FROM mn
```

No message is issued for devices whose online state has not changed.

ACTION = *REPORT

As UPDATE; in addition the device error state is logged via message EXC0857 if no volume is recognized as online.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	NKV0001	Syntax error
	64	NKV0004	Command partially processed
	64	NKV0005	Command not processed for an object
	64	NKV0006	Command not processed
	130	NKVD002	Disk monitor not available

CHECK-FILE-CONSISTENCY

Check NK-ISAM file for consistency

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING

Function

The command CHECK-FILE-CONSISTENCY checks an ISAM file that is of type BLOCK-CONTROL-INFO = *WITHIN-DATA-BLOCK and has not been correctly closed to ensure that it is consistent. This command is only meaningful for NK-ISAM files which are processed with WRITE-IMMEDIATE=*YES.

Only files identified as open (see the STATUS=*PAR(REPAIR-NEEDED=*YES) option in the SHOW-FILE-ATTRIBUTES command) are selected. The file lock is removed. Concurrent copy locks remain in place if the concurrent copy session has not yet ended. If a file is identified as open, the LLP (last logical page) is set to the last PAM page used.

If the file is on a mirrored disk (see the “DRV” manual [11]), the consistency (contents) of file blocks is restored if required.

If the file contains secondary keys, the secondary index is checked to ensure that it was created or deleted completely (since processing may have aborted at the time of creating or deleting the secondary index).

Multiblocks (an ISAM data block consisting of several PAM pages) are checked for consistency, i.e. to determine whether processing was aborted when writing a multiblock. For information on “multiblocks” see the “Introductory Guide to DMS” [13].

An existing file lock can also be removed by using the command REMOVE-FILE-ALLOCATION-LOCKS. But no inconsistencies are detected in the process. Inconsistencies in a file can be eliminated with the REPAIR-DISK-FILES command.

Format

CHECK-FILE-CONSISTENCY
FILE-NAME = <filename 1..54> / <partial-filename 2..53>
SELECT = <u>*ANY-DISK</u> / *PRIVATE-DISK / *PUBLIC-DISK / *NET-STORAGE

Operands

FILE-NAME = <filename 1..54> / <partial-filename 2..53>

Names of the files to be checked. Write access authorization is required for files which are not under the user's own user ID.

SELECT = *ANY-DISK / *PRIVATE-DISK / *PUBLIC-DISK

Restricts the files which are to be checked to the specified volume type. This operand is of use only if a partially qualified name, or the name of a file generation group, is specified for FILE-NAME.

SELECT = *ANY-DISK

Any files can be checked, irrespective of their volume types.

SELECT = *PRIVATE-DISK

Checks files which are stored on private disks.

SELECT = *PUBLIC-DISK

Checks files which are stored on public disks.

SELECT = *NET-STORAGE

Checks files which are stored on a Net-Storage volume.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed
	0	CMD0001	No action required
	0	DMS06E6	File specified in command is empty
	0	DMS06E9	No valid record found during reconstruction of ISAM file
	0	DMS06ED	Error on writing unrecoverable blocks to scratch file
	1	CMD0202	Syntactical or semantic error in command
	32	DMS0584	A state that does not allow the function to continue was reported during processing
	64	CMD0216	Privileges error
	64	DMS0501	Requested catalog not available
	64	DMS0512	Requested catalog not available

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	64	DMS051B	Requested user ID not in pubset
	64	DMS051C	User not authorized to access pubset
	64	DMS0533	Requested file not cataloged in pubset
	64	DMS0535	Specified file not shareable
	64	DMS055C	Requested file not cataloged in pubset
	64	DMS0583	An error occurred when reconstructing the file
	64	DMS0585	Error detected during catalog processing or multihost processing
	64	DMS0586	Currently not possible to access or reserve volume
	64	DMS0587	Use of specified command restricted by system administrator
	64	DMS0588	It was not possible to allocate disk space
	64	DMS05F8	It was not possible to allocate disk space Guaranteed message: DMS05F8
	64	DMS05FC	Specified user ID not in HOME pubset
	64	DMS0609	No access to system file
	64	DMS0609	No file corresponds to specified selection criterion
	64	DMS06CC	No file corresponding to specified operands
	64	DMS06E4	Reconstruction for this file type not possible with specified command
	64	DMS06FF	BCAM connection severed
	64	DMS06FF	File created with WROUT=NO. Consistency check is not practical
	130	DMS0524	System address space full
	130	DMS053C	No space in pubset catalog file
	130	DMS0582	File is currently locked or in use and cannot be processed
	130	DMS0585	Error detected during catalog processing or multihost processing
	130	DMS0586	Currently not possible to access or reserve volume
	130	DMS0588	It was not possible to allocate disk space
	130	DMS0594	Not enough virtual memory available
	130	DMS05C8	Maximum permitted number of files reached

CHECK-IMPORT-DISK-FILE

Check file import procedure

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE FILE-GENERATION-GROUP
Privileges:	TSOS STD-PROCESSING

Function

Simulates the effect of executing the IMPORT-FILE command (operand SUPPORT=DISK), but does not actually import the file, i.e the user receives a SYSOUT/SYSLST log which shows how the IMPORT-FILE command would have been processed.

The log provides a list of the files on the volume identified in the VOLUME operand, and the informative data/messages which would have been returned by actually executing the IMPORT-FILE command.

For files on the private disk or the Net-Storage volume, DMS will not check at this point whether there are file locks or protection attributes which might prevent importation. When the importation is actually performed, the user must ensure that the files are not locked, and that write access is permitted.

Privileged functions

As with the IMPORT-FILE command, systems support (TSOS privilege) can use the NEW-USER option to control which user ID the files are cataloged under.

By default, systems support (TSOS privilege) is a co-owner of all the files (and can therefore create files under all user logons). When SECOS is used, this co-ownership can be restricted for permanent files.

Systems support can generate files under any user ID (TSOS privilege). In conjunction with the SECOS software product a user can allow other user IDs to act as co-owners. Co-owners of a user ID are then also allowed to import files under that ID.

Format

CHECK-IMPORT-DISK-FILE

```

VOLUME = <alphanum-name 1..6>
,DEVICE-TYPE = <device>
,FILE-NAME = *ALL / <partial-filename 2..50 without-cat> / <filename 1..51 without-cat>
,TYPE-OF-FILE = [ *ANY ](...)
    [ *ANY ](...)
        | GENERATIONS = *YES / *NO
,OUTPUT = *SYSLST / *SYSOUT / *ALL
,NEW-USER = *NONE / <name 1..8>
,PUBSET = *STD / <cat-id 1..4>

```

Operands

VOLUME = <alphanum-name 1..6>

Specifies the VSN (volume serial number) of the volume which records the files whose importation is to be checked.

DEVICE-TYPE = <device>

Identifies the type of device on which the private disk is to be mounted.

Only device types known in the system are accepted. In interactive mode, DEVICE-TYPE=? calls up a list of the available device types.

The volume type NETSTOR must be specified for Net-Storage volumes.

Every specification of a disk device type known in the system is handled like the STDDISK specification.

FILE-NAME = *ALL / <partial-filename 2..50 without-cat> / <filename 1..51 without-cat>

Identifies the files, file generation groups or file generations for which the importation (cataloging) is to be checked.

By default, DMS will check the cataloging of all files etc. which are stored on the volume specified in the VOLUME operand under the user ID for the job which is currently running.

TYPE-OF-FILE = *ANY(...)

For file generation groups: specifies how file generation groups are to be treated.

GENERATIONS = *YES / *NO

Specifies whether the only check that is required is on the cataloging of the group entry, or whether the cataloging of the file generations which are stored on the same private disk is also to be checked.

GENERATIONS = *YES

If the group entry is held on the private disk, DMS will catalog this entry and all those of its generations which begin on this disk. If there is no group entry on the disk or in the user catalog, no file generations will be cataloged.

GENERATIONS = *NO

DMS transfers just the group entry for the FGG to the catalog.

OUTPUT = *SYSLST / *SYSOUT / *ALL

Specifies how the processing of commands is to be logged (see the [table "CHECK-IMPORT-DISK-FILE: Special return codes and messages" on page 2-286](#)).

OUTPUT = *SYSLST

The execution of commands will be logged on SYSLST; this log contains the special return codes and message texts (see [table "CHECK-IMPORT-DISK-FILE: Special return codes and messages" on page 2-286](#)).

OUTPUT = *SYSOUT

The execution of commands will be logged on SYSOUT; only the special return codes and file names are output in this log (see [table "CHECK-IMPORT-DISK-FILE: Special return codes and messages" on page 2-286](#)).

OUTPUT = *ALL

The processing of commands will be logged on SYSLST and SYSOUT (see OUTPUT=*SYSLST and OUTPUT=*SYSOUT).

NEW-USER = *NONE / <name 1..8>

User ID under which the file is to be cataloged (see also the IMPORT-FILE command).

PUBSET = *STD / <cat-id 1..4>

Specifies the pubset on which the files are to be cataloged. If a Net-Storage volume is specified in the VOLUME operand, the pubset to which the Net-Storage volume is allocated must be specified here.

PUBSET = *STD

If no specification is provided for the PUBSET, the catalog entries will be set up under the default catalog ID for the user ID (see the SHOW-USER-ATTRIBUTES command).

PUBSET = <cat-id 1..4>

Specifies the pubset on which the files are to be cataloged.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed
	1	CMD0202	Syntactical or semantic error in command
	32	DMS0584	A state that does not allow the function to continue was reported during processing
	64	CMD0216	Privileges error
	64	DMS0501	Requested catalog not available
	64	DMS0512	Requested catalog not available
	64	DMS051B	Requested user ID not in pubset Guaranteed message: DMS051B
	64	DMS051C	User not authorized to access pubset Guaranteed message: DMS051C
	64	DMS0533	Requested file not cataloged in pubset Guaranteed message: DMS0533
	64	DMS0535	Specified file not shareable
	64	DMS053E	File already cataloged on private volume
	64	DMS055C	Catalog entry not found on specified private disk
	64	DMS05FC	Specified user ID not in HOME pubset
	64	DMS05FE	Requested file(s) not found
	64	DMS0585	Error detected during catalog processing or multihost processing
	64	DMS0586	Currently not possible to access or reserve volume
	130	DMS0524	System address space full
	130	DMS0582	File is currently locked or in use and cannot be processed
	130	DMS0585	Error detected during catalog processing or multihost processing
	130	DMS0586	Currently not possible to access or reserve volume
	130	DMS0594	Not enough virtual memory available

Special return codes and messages

Code on SYSOUT SYSOUT	Message on SYSLST	Meaning
0	FILE DID NOT EXIST	There was no file with the same name and a new catalog entry would be created.
2	FILE EXISTS	A file of the same name already exists.
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 when accessing the F1 label of the private disk or the catalog of the Net-Storage volume.
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 exists for the specified disk
A	INVALID FILENAME	The path name of the file to be imported (including catid and userid) is longer than 54 characters. Pfadname = :catid: .userid:dateiname It should be noted that the file name is made up from the details specified in two <i>different</i> operands in the command: the file name is specified in the FILE-NAME operand, the catalog ID in the PUBLIC-VOLUME-SET operand For a 1-character catalog ID, the file name must not be longer than 51 characters. For a 4-character catalog ID, the file name must not be longer than 48 characters.

Table 31: CHECK-IMPORT-DISK-FILE: Special return codes and messages

Example

Check import for a private disk

```

/check-imp-disk-file vol=work01,dev-type=d3490-30,
                    file-name=*all,output=*sysout _____ (1)
% 0 :20SG:$USER1.MAX.FILE.1
% 0 :20SG:$USER1.MAX.FILE.2
% 0 :20SG:$USER1.MAX.FILE.3
% 0 :20SG:$USER1.MAX.FILE.4
% 2 :20SG:$USER1.MAX.LIB
/show-file-attr max.lib _____ (2)
%          3 :20SG:$USER1.MAX.LIB
%:20SG: PUBLIC:      1 FILE RES=      3 FRE=      3 REL=      3 PAGES
/mod-file-attr max.lib,new-name=max.lib.old _____ (3)
/check-imp-disk-file vol=work01,dev-type=d3490-30,      file-
name=*all,output=*sysout _____ (4)
% 0 :20SG:$USER1.MAX.FILE.1
% 0 :20SG:$USER1.MAX.FILE.2
% 0 :20SG:$USER1.MAX.FILE.3
% 0 :20SG:$USER1.MAX.FILE.4
% 0 :20SG:$USER1.MAX.LIB

```

- (1) The command checks whether all files of the user *USER1* on the private disk *WORK01* can be imported. The information returned by CHECK-IMPORT-DISK-FILE is to be output to SYSOUT.
- (2) The code returned for the file *MAX.LIB* is 2, i.e. there is already an existing catalog entry for the file. The command SHOW-FILE-ATTRIBUTES returns information on the existing catalog entry for the file *MAX.LIB*. The file is stored on public disk.
- (3) The command MODIFY-FILE-ATTRIBUTES is used to rename the file *MAX.LIB* to *MAX.LIB.OLD*.
- (4) A repeated CHECK-IMPORT-DISK-FILE command (see Point 1) indicates that all files of the private disk *WORK01* (including *MAX.LIB*) can be imported.

CHECK-PUBSET-MIRRORS

Check homogeneity of pubset mirroring

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS

Function

The CHECK-PUBSET-MIRRORS command checks whether mirroring of a pubset is homogeneous. The homogeneity check is performed in relation to SRDF and TimeFinder/Mirror mirroring. In the case of a pubset which is not mirrored a check is also made to see whether only individual disks are mirrored. Mirroring of the pubset is homogeneous if all volumes of the pubset have identical mirroring properties.

The homogeneity check is performed for additional mirror units (SRDF and TimeFinder/Mirror functions on Symmetrix disk storage systems) and for clones (EC-Clone functions on ETERNUS DX, TimeFinder/Clone on Symmetrix and Snapview clone on CLARiiON disk storage systems).

Format

CHECK-PUBSET-MIRRORS

PUBSET = <cat-id 1..4>

Operands

PUBSET = <cat-id 1..4>

Catalog ID of the pubset for which homogeneity of pubset mirroring (SRDF and TimeFinder/Mirror mirroring) is to be checked.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed (mirroring is homogeneous)
5	32	DMS136E	Unexpected return code when SHC-OSD was called
	32	DMS136F	Unexpected return code when a system function was called
	64	CMD0216	No authorization
	64	DMS136D	Pubset mirroring is not homogeneous
1	64	DMS136E	Specified pubset does not exist
2	64	DMS136E	Specified pubset cannot be accessed
3	64	DMS136E	SHC-OSD subsystem not available
4	64	DMS136E	Specification of a volume set not allowed
	64	DMS13DB	Homogeneity check aborted because VSN not unique

CHECK-SNAPSET-CONFIGURATION

Check and update Snapset configuration

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	SNAPSET MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS HSMS-ADMINISTRATION

Function

This command checks the Snapset configuration of an imported pubset. If Snapsets are detected which could not be connected when the pubset was imported, an attempt is made to connect these Snapsets retroactively.

Format

CHECK-SNAPSET-CONFIGURATION

PUBSET = * <u>DEFAULT-PUBSET</u> / <cat-id 1..4>

Operands

PUBSET = <cat-id 1..4>

Determines the pubset whose Snapset configuration is to be checked.
The default value is the catalog ID of the calling user's default pubset.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed without errors
	32	CMD0216	No authorization to issue command
	64	DMS1351	Internal error
	64	DMS1386	Error in the memory request
	64	DMS1389	Error in MSCF communication
	64	DMS138C	Pubset not accessible
	64	DMS13D5	No Snapset available
1	64	DMS13D7	Internal error in Snapset management: Return code of GCF
6	64	DMS13D7	Internal error in Snapset management: Return code of SHC-OSD
7	64	DMS13D7	Internal error in Snapset management: Return code of CCOPY
	64	DMS13DF	SHC-OSD subsystem not available
	64	DMS13E5	No Snapset catalog available
	64	DMS13FA	Shared pubset and Snapsets are not supported on master computer
	64	DMS148F	GCF not loaded

CHECK-TAPE-MOUNT

Check mount state of tape and MTC devices

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Device management
Domain:	DEVICE
Privileges:	OPERATING
Routing code:	T

Function

This command checks the mount state of tape devices and magnetic tape cartridge devices. The tape monitor (TM) is called to update the online state for the specified devices. A typical application is where the mounting or dismounting of a tape or magnetic tape cartridge is not recognized by the system because of the absence of an activation interrupt.

Format

CHECK-TAPE-MOUNT
UNIT = *ALL / list-poss(10): <alphanum-name 2..2> / <alphanum-name 4..4> ,ACTION = *UPDATE / *REPORT

Operands

UNIT =

Defines the devices whose mount state is to be checked.

UNIT = *ALL

All tape devices having the following attributes are checked:

- ATTACHED or DETACH-PENDING
- ACTION-STATE = NO ACTION
- PHASE ≠ MOUNT and PHASE ≠ IN-USE.

UNIT = list-poss(10): <alphanum-name 2..2> / <alphanum-name 4..4>

The devices whose mount state is to be checked and updated are defined by means of their mnemonic name. A maximum of 10 devices may be specified.

ACTION =

Defines the logging of the online state.

ACTION = *UPDATE

An updated online state is to be logged by the following messages (default value):

```
NKVT010 TAPE vsn MOUNTED ON DEVICE mn
NKVT011 TAPE vsn DISMOUNTED FROM DEVICE mn
```

No message is issued for devices whose online state has not changed.

ACTION = *REPORT

Logging as for UPDATE; in addition the device error state is logged via message EXC0858 if no volume is recognized as online.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	No error
	1	NKV0001	Syntax error
	64	NKV0004	Command partially processed
	64	NKV0005	Command not processed for an object
	64	NKV0006	Command not processed
	130	NKVT002	Tape monitor not available

CLOSE-VARIABLE-CONTAINER

Close variable container

Description status:	SDF-P-BASYS V2.5E
Functional area:	Procedures
Domain:	PROCEDURE
Privileges:	STD-PROCESSING OPERATING HARDWARE-MAINTENANCE SECURITY-ADMINISTRATION SAT-FILE-MANAGEMENT SAT-FILE-EVALUATION

Function

The CLOSE-VARIABLE-CONTAINER command closes the specified variable containers.

Note

If a variable container is closed before its scope ceases to exist, the variables remain declared but can no longer be accessed. All access attempts are rejected with error message SDP1030.

Format

CLOSE-VARIABLE-CONTAINER

CONTAINER-NAME = <composed-name 1..64 with-wild(80)> / list-poss(2000): <composed-name 1..64>
--

Operands

CONTAINER-NAME =

Name of the variable container.

CONTAINER-NAME = <composed-name 1..64 with-wild(80)>

Variable container whose name corresponds to the pattern string specified.

CONTAINER-NAME = list-poss(2000): <composed-name 1..64>

List of variable container names.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD0202	Syntax error
	3	CMD2203	Incorrect syntax file
	32	CMD0221	System error (internal error)
	64	CMD0216	User does not have required privilege
	64	SDP0091	Semantic error
	130	SDP0099	No more address space available

Example

See the SHOW-VARIABLE-CONTAINER-ATTR command in the “SDF-P” manual [[34](#)].

CLOSE-VIRTUAL-DEVICE-DIALOG

Close dialog with virtual printer

Description status:	SPOOL V4.9A
Functional area:	Controlling spoolout job
Domain:	SPOOL-PRINT-SERVICES
Privileges:	STD-PROCESSING PRINT-SERVICE-ADMINISTRATION

Function

This command closes the dialog between the virtual device and the application procedure in which the call takes place. It is executed in batch mode only.

The command is part of a set of four commands which enable an application to be created in the form of an S procedure (for an example, see the OPEN-VIRTUAL-DEVICE-DIALOG command). These commands manage the dialog between a virtual device and the application which was started in batch mode as an S procedure:

- OPEN-VIRTUAL-DEVICE-DIALOG
- GET-JOB-FROM-VIRTUAL-DEVICE
- RETURN-JOB-TO-VIRTUAL-DEVICE
- CLOSE-VIRTUAL-DEVICE-DIALOG

Format

CLOSE-VIRTUAL-DEVICE-DIALOG

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error; command successfully processed
	32	SCP0974	Unexpected command
5	32	SCP0974	Memory request error

Notes

1. If the dialog cannot be established, the command is rejected and a return code is set.
2. If no initialization took place (OPEN-VIRTUAL-DEVICE-DIALOG command), the command is rejected and a return code is set.
3. This command can only be used in batch mode.
4. Spin-off processing is activated each time an error is detected.

COMPARE-DISK-FILES

Compare disk files

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING TSOS SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT HARDWARE-MAINTENANCE

Function

The COMPARE-DISK-FILES command compares two disk files block by block (UPAM) or record by record (SAM, ISAM) and informs the user of the result of the comparison.

The files can reside on public volumes, Net-Storage volumes or private disks. Permanent files, temporary files or work files can be compared.

Entire file generation groups cannot be compared, although individual file generations can.

The files to be compared must be identical with respect to the following properties:

- Access method (FILE-STRUCTURE)
- Block format (BLOCK-CONTROL-INFO)
The default BLOCK-CONTROL-INFO=*IGNORE-ATTRIBUTE also permits different block formats.
- Coded character set (CODED-CHARACTER-SET)
- For SAM files:
 - REC-SIZE when REC-FORM=F
- For ISAM files:
 - REC-SIZE when REC-FORM=F
 - Properties of the ISAM key (KEY-LEN, KEY-POS, LOG-FL-LEN, VAL-FL-LEN)
 - Properties of the secondary key (KEY-LEN, KEY-POS, DUPKEY) for NK-ISAM
- For UPAM files:
 - Blocking factor (BUFFER-LENGTH)
 - Number of occupied PAM pages (HIGHEST-USED-PAGE)

The command will be rejected for files with one of the following properties:

- The file is empty.
- The file is open.
- The file is locked (e.g. SECURE lock)
- The REPAIR-NEEDED or NO-DMS-ACCESS label is set.

PLAM libraries are UPAM files and are compared block by block. The utility routine "LMS" [21] permits a comparison at member level.

The TPCOMP2 utility routine is available to compare tape files, see the "Utility Routines" manual [9].

Format

COMPARE-DISK-FILES

FILE-NAME1 = <filename 1..54>

,**FILE-NAME2** = <filename 1..54>

,**BLOCK-CONTROL-INFO** = *IGNORE-ATTRIBUTE / *INCLUDE-ATTRIBUTE /

,**PAMKEY-USER-INFO** = *INCLUDE / *IGNORE

Operands

FILE-NAME1 = <filename 1..54>

Name of the first file which is to be compared.

FILE-NAME2 = <filename 1..54>

Name of the second file which is to be compared.

BLOCK-CONTROL-INFO = *IGNORE-ATTRIBUTE / *INCLUDE-ATTRIBUTE

The files' block format is included in the comparison (*INCLUDE-ATTRIBUTE) or ignored (*IGNORE-ATTRIBUTE).

PAMKEY-USER-INFO = *INCLUDE / *IGNORE

*Only relevant for UPAM files with BLOCK-CONTROL-INFO=*PAMKEY.*

The user information in the PAM key is included in the comparison (*INCLUDE) or ignored (*IGNORE).

Output information about the file comparison

If the files are identical, message DMS0630 is issued.

If the files are not identical, message DMS0631 is issued. Depending on the file type, further information is output:

- For SAM files:
Message DMS0632 displays the number of the record in which the first difference occurs. The block and record numbers within the data block are also displayed for this position.
- For ISAM files:
Message DMS0633 displays the number of the record in which the first difference occurs.
- For UPAM files:
Message DMS0634 displays the number of the 2K data block and the byte number within this data block in which the first difference occurs.
If PAMKEY-USER-INFO=*INCLUDE is used and the first difference in the user information is found in the PAM key, message DMS0635 displays the number of the associated 2K data block.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	DMS0630	Command executed without error. The files are identical.
	0	DMS0631	Command executed without errors. The files are not identical.
	1	CMD0202	Syntax or semantic error in file name
	32	DMS05C7	Unexpected internal error in DMS
	64	CMD0102	Interrupted by K2 key
	64	CMD0216	Command privilege missing
	64	DMS0501	File catalog not available
	64	DMS0512	File catalog not found
	64	DMS051B	User ID not in pubset
	64	DMS051C	User not authorized to access pubset
	64	DMS0535	Specified file not shareable
	64	DMS0585	An error was detected during catalog processing or during multiprocessor processing
	64	DMS05FC	User ID not in home pubset
	64	DMS0636	File attributes do not permit a comparison
	64	DMS0684	File does not exist
	130	DMS0524	System address space full
	130	DMS0582	File is currently locked or in use and cannot be processed
	130	DMS0594	Not enough virtual memory available

CONCATENATE-DISK-FILES

Join SAM files together

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING TSOS

Function

The CONCATENATE-DISK-FILES command joins two or more SAM files together, concatenating them to form a single file. This is, however, conditional on the files all having matching FILE-STRUCTURE, BUFFER-LENGTH, RECORD-FORMAT, BLOCK-CONTROL-INFORMATION and CODED-CHARACTER-SET structure attributes. Files with a fixed record length (RECORD-FORMAT=FIXED) must have matching record lengths (RECORD-SIZE).

The target file is freely selectable, but by default the first of the files being concatenated is used as the target file. The command is rejected if the user does not have write permission for the target file.

Methodology

The files are copied record by record in the order specified in the FROM-FILES operand, with each appended file beginning in a separate data block. If an error (e.g. no read permission) occurs while copying is in progress, processing of the command aborts with an error code. If the target file (TO-FILE operand) has already been created, it will not be deleted.

A message provides information about the file which could not be processed.

Processing of encrypted files is possible only if the relevant crypto passwords are entered in the task's crypto password table (see the ADD-CRYPTO-PASSWORD command).

Format

CONCATENATE-DISK-FILES

FROM-FILES = list-poss(255): <filename 1..54>

TO-FILE = *FIRST-INPUT-FILE / <filename 1..54>

Operands

FROM-FILES = list-poss(255): <filename 1..54>

Defines the list of files which are to be copied, in the given order, to the file specified in the TO-FILE operand.

TO-FILE = *FIRST-INPUT-FILE / <filename 1..54>

Specifies the file to which the files specified in the FROM-FILE operand are to be copied. The command is rejected if the user does not have write permission for the target file.

TO-FILE = *FIRST-INPUT-FILE

The files specified in the FROM-FILES operand are to be copied to the first file listed in the FROM-FILES operand. In other words, the files following the first file are to be appended to the first file in the order specified in the FROM-FILES operand.

TO-FILE = <filename 1..54>

Specifies the name of the file to which the files specified in the FROM-FILES operand are to be copied. If a file with this name already exists, it will be overwritten. If the file is one of the set specified in the FROM-FILES operand, it must be the first file listed there.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	CMD0202	Syntactical/semantic error in file name
	64	DMS0681	DMS error during processing. For more detailed information the inserts can be queried using the chargeable product SDF-P

Example

```

/ass-syslst to=text.1,sys-num=1 ----- (1)
/ass-syslst to=text.2,sys-num=2
/ass-syslst to=text.3,sys-num=3

/wr-text `We begin with TEXT.1 ... ` ,output=*syslst(1) ----- (2)
/wr-text ` .... then comes TEXT.2 ... ` ,output=*syslst(2)
/wr-text ` .... and last there is TEXT.3! ` ,output=*syslst(3)

/ass-syslst *primary,sys-num=1 ----- (3)
/ass-syslst *primary,sys-num=2
/ass-syslst *primary,sys-num=3

/concat-disk-file from=(text.1,text.2,text.2,text.2,text.3),to=text.all ----- (4)

```

/sh-file text.all (5)

```

We begin with TEXT.1 ...
.... then comes TEXT.2 ...
.... then comes TEXT.2 ...
.... then comes TEXT.2 ...
.... and last there is TEXT.3!

% SH00301 WARNING: END OF FILE REACHED
                                     S*S0F+   1(   1)
e
% SH00500 :10SN:$USERXY01.TEXT.ALL CLOSED

```

- (1) The SYSLST system file (SYSLST01, SYSLST02 and SYSLST03) is assigned to files TEXT.1, TEXT.2 and TEXT.3.
- (2) Using the WRITE-TEXT command, text is written to SYSLST01, SYSLST02 and SYSLST03 and hence to the assigned files TEXT.1, TEXT.2 and TEXT.3.
- (3) The system file assignment is canceled. As a result TEXT.1, TEXT.2 and TEXT.3 are closed and made accessible.
- (4) The CONCATENATE-DISK-FILES command is used to copy the contents of TEXT.1 once, TEXT.2 three times and TEXT.3 once to a file named TEXT.ALL.
- (5) The SHOW-FILE command displays the contents of TEXT.ALL. Output ends with statement "e" (END).

CONNECT-CMD-SERVER

Create/modify entry in operator command table

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Operator function control
Domain:	not allocated
Privileges:	OPERATING
Routing code:	K

Function

The CONNECT-CMD-SERVER creates or modifies an entry in the command table. This entry links the command specified by means of CMD-NAME with a \$CONSOLE application, which then functions as the command server.



The CONNECT-CMD-SERVER is only permitted for \$CONSOLE applications.

Format

CONNECT-CMD-SERVER

```

CMD = *PARAMETERS (...)
  *PARAMETERS(...)
    |
    | CMD-NAME = <structured-name 1..30>
    |
    | ,SAME-NAME = *NONE / list-poss(8): <structured-name 1..30>
    |
    | ,AUTHORIZATION-CODE = *STD / <c-string 1..1>
    |
    | ,PASSWORD-POSSIBLE = *NO / *YES
    |
  ,SELECT = *EXTERNAL-SERVER (...)
    *EXTERNAL-SERVER(...)
      |
      | APPLICATION-NAME = *OWN
      |
      | ,COMPLETION-CONTROL = *NO / *YES(...)
      |
      | *YES(...)
      |
      |   QUESTION-POSSIBLE = *YES / *NO

```


Operands

CMD = *PARAMETERS(...)

Defines a structure for the command for which an entry is to be created or modified.

CMD-NAME = <text 1..30>

Defines the full command name for which an entry in the command table is to be created or modified. The command name may be up to 30 characters long and must comply with the naming convention for command names.

SAME-NAME =

Defines aliases.

This operand is only relevant if a command is being defined. This operand is ignored when changing the attributes of commands that have already been entered.

SAME-NAME = *NONE

The command may only be entered using the command name specified using CMD-NAME.

SAME-NAME = list-poss(8): <text 1..30>

Defines the set of aliases that may be used in place of the command name defined using CMD-NAME. An alias may be up to 30 characters long and must comply with the naming convention for command names.

AUTHORIZATION-CODE =

Defines the authorization code which a command issuer must have in order to use the command identified by CMD-NAME.

An authorization code can only be generated when a new entry in the command table is being created. This operand is ignored when an existing entry is being modified.

AUTHORIZATION-CODE = *STD

This operand causes the specified command to be protected by the authorization code (default = E) defined in the class 2 system parameter NBACODE.

AUTHORIZATION-CODE = <c-string 1..1>

Explicit specification of an authorization code (A..Z, 0-9 or *,#, @, \$).

PASSWORD-POSSIBLE =

Specifies whether SDF is to be called and passwords deleted before the command is logged in the CONSLOG file.

PASSWORD-POSSIBLE = *NO

The command is not checked to see whether it contains a password before being entered in the CONSLOG file.

PASSWORD-POSSIBLE = *YES

Causes SDF to be called in order to delete any passwords from the command before it is logged in the CONSLOG file.

This operand value is only to be used in connection with SDF commands whose operands are all explicitly defined in the SDF syntax file (syntax of each operand can be analyzed).

Passwords in special commands that are not contained in the SDF syntax file or whose operands are not explicitly defined will be included in the system CONSLOG file.

SELECT = *EXTERNAL-SERVER(...)

The entry should refer to a special operator command. Special operator commands are commands which are executed in \$CONSOLE applications (authorized user programs).

APPLICATION-NAME =

Defines the name of the \$CONSOLE application.

APPLICATION-NAME = *OWN

The entry is to be made for a special operator command for which the command issuer should also function as the command processor.

COMPLETION-CONTROL =

Specifies whether commands to be executed in \$CONSOLE applications can continue to be administered by the OPR subsystem once they have been passed to the application. This is always the case if the \$CONSOLE application executing the command informs the OPR subsystem that command processing has terminated.

COMPLETION-CONTROL = *NO

The OPR subsystem considers the command involved to be terminated as soon as it has been passed to the executing application. Once the command has been passed on successfully, the OPR subsystem issues message NBR0768 to indicate command termination.

COMPLETION-CONTROL = *YES(...)

The OPR subsystem does not consider the command involved to be terminated until it has received a command termination acknowledgment from the executing application. The OPR subsystem indicates command termination, as for normal operator commands, by issuing message NBR0740.

QUESTION-POSSIBLE =

Informs the system whether the application may ask questions regarding the command during command execution.

QUESTION-POSSIBLE = *YES

The application may ask questions regarding the command during command execution. In this case, the job ID must be unique within the set of commands being processed by this application. Otherwise, the command is rejected.

QUESTIONS-POSSIBLE = *NO

The application does not ask any questions regarding the command during command execution.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
2	0	NBR1125	Parameters only evaluated at initial logon
2	0	NBR1118	Command executed, but at least one SAME name rejected
	1	CMD0202	Syntax error
	64	NBR1113	Maximum number of links already reached
1	64	NBR1115	Semantic operand error
	64	NBR1119	Command permitted only for \$CONSOLE application
	64	NBR1124	Operand error connected with selected interface version

CONNECT-GS-SERVER

Establish connection to the GS units in a complex

Description status:	GSMAN V19.0A
Functional area:	Global storage administration
Domain:	DEVICE
Privileges:	TSOS

Function

Systems support can use the CONNECT-GS-SERVER command to establish a connection between the local GS server (local host) and the GS units of a GS complex during a reconfiguration. The GS server must form part of the GS complex, i.e. it must be online. The command can be used in the following cases:

- There is as yet no connection in the GS complex and the local GS server is to be connected with all the GS units in the GS complex.
- The GS units are already connected to another GS server in the GS complex.

In a parallel HIPLEX, the command is only accepted if the XCS network currently consists of only one node.

In a VM2000 guest system which has assigned GS as real and VM as shared, the CONNECT-GS-SERVER command results in the hardware connection of the GS server provided that this has not already been connected by another guest system. Otherwise the guest system issuing the command is only virtually connected.

The command calls background functions which may take some time to process at the relevant SVPs. Other BS2000 functions which need the SVP (e.g. IOCCOPY or CPU and channel reconfiguration commands) should not be called at the same time. If possible, the BS2000 load should be reduced before calling the command.

For more details on managing global storage see the “Introduction to System Administration” [14].

Systems support can use the SHOW-GS-COMPLEX-CONFIGURATION command to obtain information on the current configuration of the GS complex.

Format

CONNECT-GS-SERVER

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD0202	Syntax error
	32	CMD0221	Internal error
	64	CMD0216	Privileges error
	64	EGC0005	Command cancelled by user
	64	EGC2014	SVP error
	64	EGC2210	GS server not in GS complex
	64	EGC2211	GS server already connected
	64	EGC2215	XCS not yet ready
	64	EGC2216	Other active GS servers in XCS
	128	EGC0010	GSMAN subsystem is not ready
	128	EGC2221	System error

CONSOLE

Assign standby operator terminals

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Operator terminal control
Domain:	not allocated
Privileges:	OPERATING
Routing code:	*



If the “Operator LOGON” function is used (system parameter NBCONOPI=Y), the command is not available.

In the mode without operator LOGON (NBCONOPI=N), the command is available with its BS2000/OSD-BC V2.0 functionality.

Function

The CONSOLE command changes the assignment of standby operator terminals or switches over to standby operator terminals. It also serves to cancel assignments and switchover operations.

Restrictions

The CONSOLE command can be issued from the main operator terminal only. (Any modification of the routing code '*' for the command has no effect.)

The command is available in ISP format only; in other words, there are no SDF functions (such as syntax analysis or help).

Format

Operation	Operands
$\left. \begin{array}{l} \{ \text{CONSOLE} \\ \{ \text{CON} \} \end{array} \right\}$	$\left. \begin{array}{l} \left\{ \begin{array}{l} \text{D[EFINE]}, \left\{ \begin{array}{l} \text{R[REPLACEMENT]}=(mn_1, mn_2[, mn_1, mn_2, \dots]) \\ \text{O[RIGINAL]}=\left\{ \begin{array}{l} mn \\ (mn, mn, \dots) \\ \text{ALL} \end{array} \right\} \end{array} \right\} \\ \\ \left\{ \begin{array}{l} \text{S[WITCH]}, \left\{ \begin{array}{l} \text{OFF}=\left\{ \begin{array}{l} mn \\ (mn, mn, \dots) \end{array} \right\} \\ \text{ON}=\left\{ \begin{array}{l} mn \\ (mn, mn, \dots) \end{array} \right\} \end{array} \right\} \\ \\ \text{H[ELP]}[, \text{CS}=\left\{ \begin{array}{l} mn \\ (mn, mn, \dots) \\ \text{ALL} \end{array} \right\}] \end{array} \right\}$

Operands

DEFINE Changes the assignment of standby operator terminals.

REPLACEMENT

=(mn₁,mn₂[,mn₁,mn₂,...])

Operator terminal “mn₁” is assigned standby operator terminal “mn₂”. If more than 2 operator terminals are specified, the assignments are effected in pairs and in sequence.

“mn” is the mnemonic device name of an operator terminal.

Note

TELESERVICE buffers cannot be assigned as standby operator terminals.

ORIGINAL

=mn

=(mn,mn)

The operator terminals “mn” are assigned the standby operator terminals defined in the OPR parameter service.

=ALL

All operator terminals are assigned the standby operator terminals defined in the OPR parameter service.

If no operator terminal is specified, the main operator terminal is assigned the standby operator terminal defined in the OPR parameter service.

SWITCH

Switches operator terminals over to their standby operator terminals or resets this switchover.

OFF

=mn

=(mn,mn,...)

The operator terminals “mn” are switched over to their standby operator terminals.

If only one operator terminal is available to the system, the command is rejected.

The main operator terminal is not permitted to switch itself off.

ON

=mn

=(mn,mn,...)

The operator terminals “mn” regain their functions. The standby operator terminals are stripped of the functions of the reactivated operator terminals.

If the former main operator terminal is reactivated, it regains only its previous routing codes; the “main operator terminal” function is retained by the standby operator terminal.

HELP

Provides information on the allocation of standby operator terminals defined in the OPR parameter service or by the CONSOLE command.

CS

=mn

=(mn,mn,...)

Outputs the standby operator terminal allocation for the operator terminals “mn”.

=ALL

Outputs the standby operator terminal allocation for all operator terminals.

If no operator terminal is specified, the standby operator terminal allocation of the main operator terminal is output.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD0202	Incorrect operand
	1	EXC0646	Incompatible operands
	1	EXC0648	Console names not specified in pairs in DEFINE operand
	64	EXC0647	Console has invalid console name
	64	EXC0649	TELESERVICE consoles may not be assigned as standby consoles
	64	EXC0654	Command may only be issued from main console
	64	NBR0710	Deactivation of main console not permitted

CONVERT-FILE-TO-PDF

Convert a text file to a PDF file

Description status:	CONV2PDF V1.0B
Functional area:	File processing
Domain:	FILE, SPOOL-PRINT-SERVICES, UTILITIES
Privileges:	STD-PROCESSING

Function

The CONVERT-FILE-TO-PDF command converts text files (SAM or ISAM files) to PDF files. It is used to print-edit BS2000 text files and library elements for use on a PC where they can be read on a user-friendly basis or printed out page by page using a PDF viewer (e.g. Adobe Reader).

The user can also specify more than one text file in a command call. Output then takes place either to individual PDF files or to one combined PDF file. When a combined PDF file is used, bookmarks are automatically set for the start pages of the text files which have been converted. Further parameters enable the user to control the content and layout of the PDF file:

- processing records fully or only in part
- line spacing (existing print control characters are optionally interpreted)
- page size
- character density
- margins
- font
- truncating or wrapping of long data lines
- background picture on the PDF pages

Alternatively the user can also control the layout via spool parameters, as with the PRINT-DOCUMENT command.

It must be borne in mind that depending on the layout only a particular number of characters can be displayed in a line. Data lines which extend beyond the right-hand margin are truncated or optionally wrapped when a PDF file is created.

The data in the PDF file is compressed. File transfer must be performed in binary mode.

Format

<p>CONVERT-FILE-TO-PDF</p> <p>FROM-FILE = list-poss(16): <filename 1..54 with-wild(80)> / *LIBRARY-ELEMENT(...)</p> <p>*LIBRARY-ELEMENT(...)</p> <ul style="list-style-type: none"> LIBRARY = <filename 1..54 without-vers> ,ELEMENT = <composed-name 1..64 with-under with-wild(80)>(... <ul style="list-style-type: none"> <composed-name 1..64 with-under with-wild(80)>(... <ul style="list-style-type: none"> VERSION = *HIGHEST-EXISTING / *UPPER-LIMIT / <ul style="list-style-type: none"> <composed-name 1..24 with-under with-wild(40)> ,BASE = *STD / <composed-name 1..24 with-under with-wild(40)> ,TYPE = <alphanum-name 1..8 with-wild(12)> <p>,TO-FILE = *BY-SOURCE / *CONCATENATE(...) / list-poss(16): <filename 1..54 with-wild-constr(80)>(...</p> <p>*CONCATENATE(...)</p> <ul style="list-style-type: none"> TO-FILE = <filename 1..54 without-gen> ,BOOKMARK = *BY-SOURCE / *NONE / list-poss(16): <filename 1..54 with-wild-constr(80)> <ul style="list-style-type: none"> <filename 1..54 with-wild-constr(80)>(... WITH-VERSION = *STD / *YES / *NO ,WITH-TYPE = *STD / *YES / *NO <p>,WRITE-MODE = *CREATE / *REPLACE-ONLY / *ANY</p> <p>,FILE-FORMAT = *STD / *SAM / *PAM</p> <p>,PRESENTATION = *DIRECT-PARAMETERS(...) / *SPOOL-PARAMETERS(...)</p> <p>*DIRECT-PARAMETERS(...)</p> <ul style="list-style-type: none"> RECORD-PART = *ALL / *PARAMETERS(...) *PARAMETERS(...) <ul style="list-style-type: none"> FIRST-CHARACTER = <u>1</u> / <integer 2..32767> ,LAST-CHARACTER = *STD / <integer 1..32767> ,LINE-SPACING = <u>1</u> / <u>2</u> / <u>3</u> / *BY-EBCDIC-CONTROL(...) / *BY-IBM-CONTROL(...) / <ul style="list-style-type: none"> *BY-ASA-CONTROL(...) *BY-EBCDIC-CONTROL(...) <ul style="list-style-type: none"> CONTROL-CHAR-POS = *STD / <integer 1..2040> *BY-IBM-CONTROL(...) <ul style="list-style-type: none"> CONTROL-CHAR-POS = *STD / <integer 1..2040> *BY-ASA-CONTROL(...) <ul style="list-style-type: none"> CONTROL-CHAR-POS = *STD / <integer 1..2040>
--

(Part 1 of 3)

```

,PAGE-SIZE = *A4 / *A4-LANDSCAPE / *A3 / *A3-LANDSCAPE / *A5 / *A5-LANDSCAPE /
*A6 / *A6-LANDSCAPE / *PARAMETERS(...)

*PARAMETERS(...)
    | WIDTH = <integer 2..2040>
    | ,HEIGHT = <integer 2..2040>

,MARGINS = *PARAMETERS(...)

*PARAMETERS(...)
    | LEFT = 20 / <integer 0..2040>
    | ,RIGHT = 20 / <integer 0..2040>
    | ,TOP = 20 / <integer 0..2040>
    | ,BOTTOM = 20 / <integer 0..2040>

,DENSITY = *PARAMETERS(...)

*PARAMETERS(...)
    | LINES-PER-INCH = 6 / <integer 3..24>

,FONT = *PARAMETERS(...)

*PARAMETERS(...)
    | NAME = *COURIER / *HELVETICA / *TIMES
    | ,CHARACTER-STYLE = *NORMAL / *BOLD / *ITALIC / *BOLD-ITALIC
    | ,SIZE = 8 / <integer 1..72>

,LINE-TRUNCATION = *YES / *NO

,OVERLAY = *NONE / *PARAMETERS(...)

*PARAMETERS(...)
    | FROM-FILE = <filename 1..54 without-gen-vers>
    | ,FRAME = *PAGE / *TEXT / *CUSTOM(...)
        | *CUSTOM(...)
            | LEFT = 0 / <integer 0..2040>
            | ,RIGHT = 0 / <integer 0..2040>
            | ,TOP = 0 / <integer 0..2040>
            | ,BOTTOM = 0 / <integer 0..2040>
        | ,HORIZONTAL-ALIGNMENT = *LEFT / *RIGHT / *CENTER
        | ,VERTICAL-ALIGNMENT = *TOP / *BOTTOM / *CENTER
        | ,SCALE = *UNCHANGED / *FIT-HEIGHT / *FIT-WIDTH / *FIT-FRAME

```

(Part 2 of 3)

```

*SPOOL-PARAMETERS(...)
  RECORD-PART = *ALL / *PARAMETERS(...)
    *PARAMETERS(...)
      FIRST-CHARACTER = 1 / <integer 2..32767>
      ,LAST-CHARACTER = *STD / <integer 1..32767>
    ,LINE-PER-PAGE = *STD / <integer 1..32767>
    ,LINE-SPACING = 1 / 2 / 3 / *BY-EBCDIC-CONTROL(...) / *BY-IBM-CONTROL(...) /
      *BY-ASA-CONTROL(...)
      *BY-EBCDIC-CONTROL(...)
        CONTROL-CHAR-POS = *STD / <integer 1..2040>
      *BY-IBM-CONTROL(...)
        CONTROL-CHAR-POS = *STD / <integer 1..2040>
      *BY-ASA-CONTROL(...)
        CONTROL-CHAR-POS = *STD / <integer 1..2040>
    ,FORM-NAME = *STD / <alphanum-name 1..6>
    ,LOOP-NAME = *STD / <alphanum-name 1..3>
    ,CHARACTER-SET = *STD / <alphanum-name 1..3>
    ,PRINTER-TYPE = *HP90-PRINTER / *HP-PRINTER / *LP-PRINTER
    ,LEFT-MARGIN = 20 / <integer 0..2040>
    ,ROTATION = *NO / *YES

```

(Part 3 of 3)

Operands

FROM-FILE = list-poss(16): <filename 1..54 with-wild(80)> / *LIBRARY-ELEMENT(...)

Name of the text file to be converted which exists either as a file or a library element.

Multiple files or library elements can be specified using wildcards or in a list. Only SAM or ISAM files are accepted (except for SAM files with REC-FORM=U).

FROM-FILE = *LIBRARY-ELEMENT(...)

The specified element from a PLAM library is to be converted. An element is fully defined by its name, its type and its version.

The records of an element are assigned to particular record types. There are 255 record types. A distinction is drawn between user record types (1 through 159) and special record types (160 through 255). Only the user record types of an element can be converted to PDF.

LIBRARY = <filename 1..54 without-vers>

Name of the PLAM library from which an element is to be converted.

ELEMENT = <composed-name 1..64 with-under with-wild(80)>(…)

Name of the element to be converted. Multiple elements can be output using wildcards.

**VERSION = *HIGHEST-EXISTING / *UPPER-LIMIT /
<composed-name 1..24 with-under with-wild(40)>**

The version of the element which is to be output. Default is *HIGHEST-EXISTING, i.e. the last element in alphabetical order. If the version is specified in wildcard format, and if there are library elements with the same names to which the wildcard specification applies, then all of these library elements are output.

BASE = *STD / <composed-name 1..24 with-under with-wild(40)>

Prefix for version selection. In conjunction with VERSION=*HIGHEST-EXISTING this enables the highest version to be addressed with a particular prefix.

BASE=*STD has the same effect as BASE=*

For detailed information on specifying the base, see the "LMS" manual [21].

TYPE = <alphanum-name 1..8 with-wild (12)>

The type of the library element to be output.

If specified in wildcard format, the name consists of a maximum of 12 alphanumeric characters.

Note

The records of LMS elements of type C, L or R belong to the special record types (160 through 255). That is the reason why no records of such elements can be converted.

TO-FILE =

Specifies the name of the PDF file to be created.

TO-FILE = *BY-SOURCE

The name of a created PDF file consists of the name of the corresponding text file (file or element name) and the suffix "PDF". If an illegal file name results, the command is rejected.

TO-FILE = *CONCATENATE(...)

All the text files (files or elements) specified are to be combined in one PDF file. In this case each text file starts on a new page.

TO-FILE = <filename 1..54 without-gen>

Name of the PDF file.

BOOKMARK =

Controls the setting of bookmarks in the PDF file.

BOOKMARK = *BY-SOURCE

The start page of each of the text files is assigned a bookmark with the name of the text file (file or element name).

BOOKMARK = *NONE

Suppresses the setting of bookmarks.

BOOKMARK = list-poss(16): <filename 1..54 with-wild-constr(80)>

The start page of each of the text files is assigned a bookmark with a name which is specified explicitly. When a list is specified, the names are set as bookmarks in the specified order. When a constructor string is specified, the bookmarks are mapped in accordance with the wildcard sequence specified in the FROM-FILE or ELEMENT-NAME operand.

TO-FILE = list-poss(16): <filename 1..54 with-wild-constr(80)>(…)

Specifies the name(s) of the PDF file(s) explicitly. If wildcards are specified in FROM-FILE or ELEMENT-NAME, a constructor string can be specified for the PDF files.

The default for library elements is that the specified name is also assigned a suffix containing the type and version of the element (format: <to-file>.<type>.<version>). The creation of this suffix can also be controlled using the following operands:

WITH-VERSION = *STD / *YES / *NO

Evaluated only for library elements:

Specifies whether the suffix should contain the element version.

*STD specifies *YES as the default.

WITH-TYPE = *STD / *YES / *NO

Evaluated only for library elements:

Specifies whether the suffix should contain the element type.

*STD specifies *YES as the default.

WRITE-MODE = *CREATE / *REPLACE-ONLY / *ANY

Determines the write mode for the PDF files which are to be created. The default value is *CREATE, i.e. a new file is created. The command is rejected for a file which already exists.

WRITE-MODE = *REPLACE-ONLY

The output file must already exist and is overwritten during conversion. If the file does not already exist, the command is rejected.

WRITE-MODE = *ANY

A new output file is created. If the file already exists, it is overwritten.

FILE-FORMAT = *STD / *SAM / *PAM

Determines the file format of the PDF file.



An existing PDF file can be converted retroactively to the other file format using the SAM/PAM converter (which is called using the START-SAM-PAM-CONVERTER command, see the “BS2ZIP” manual [53]).

FILE-FORMAT = *STD

Uses the file format which is defined in the SYSPAR.CONV2PDF.<version> parameter file. The parameter file is searched for at the following storage locations (search takes place in the specified order):

1. Caller’s user ID
2. TSOS user ID

If no parameter file is found, FILE-FORMAT=*SAM applies.



You will find the template of a parameter file with the name SYSPAR.CONV2PDF.<version> under the installation ID of CONV2PDF.

FILE-FORMAT = *SAM

The PDF file is created in SAM file format using REC-FORM=U.

FILE-FORMAT = *PAM

The PDF file is created in PAM file format and BLOCK-CONTROL=NO. In this file format the PDF file can also be created as a node file.

PRESENTATION = *DIRECT-PARAMETERS(...) / *SPOOL-PARAMETERS(...)

Defines the layout of the PDF file. Direct specification of the layout properties is preset. Alternatively the layout can be determined by specifying spool parameters (as with print output).

PRESENTATION = *DIRECT-PARAMETERS(...)

Specifies the layout of the PDF file directly.

RECORD-PART = *ALL / *PARAMETERS(...)

Defines whether the records of the text file are to be processed in full or if only a particular part of each record is to be processed. When the default value *ALL is specified, the records are processed in full.

RECORD-PART = *PARAMETERS(...)

Specifies which part of the records is to be processed. Only the specified part is taken into account during conversion to PDF. This entry can be used, for example, to omit the ISAM key or control characters in the PDF file.

FIRST-CHARACTER = 1 / <integer 2..32767>

Allows a byte number (record column) to be specified indicating the point as of which the records of a file are to be output. The bytes of a record are numbered consecutively from left to right starting with 1; ISAM keys and control characters are components of a record.

LAST-CHARACTER = *STD / <integer 1..32767>

Specifies the byte indicating the point at which printing of each record is to stop.

LINE-SPACING = 1 / 2 / 3 / *BY-EBCDIC-CONTROL(...) / *BY-IBM-CONTROL(...) / *BY-ASA-CONTROL(...)

Specifies the number of line feeds and the way in which control characters are interpreted.

LINE-SPACING= 1 / 2 / 3

The records are to be printed out with 1-, 2- or 3-line spacing.

LINE-SPACING = *BY-EBCDIC-CONTROL(...) / *BY-IBM-CONTROL(...) / *BY-ASA-CONTROL(...)

The contents of the first byte of each record are to be interpreted as EBCDIC, IBM or ASA feed control characters (refer to PRINT-DOCUMENT command).

CONTROL-CHAR-POS = *STD / <integer 1..2040>

Number of the data byte which is interpreted as the feed control character. In the case of records of variable length, the fields containing the length information are not counted as data.

PAGE-SIZE = *A4 / *A4-LANDSCAPE / *A3 / *A3-LANDSCAPE / *A5 / *A5-LANDSCAPE / *PARAMETERS(...)

Determines the page size of the PDF file:

Operand value	DIN format	Width x Height (mm)
*A4	DIN-A4	210 x 297
*A4-LANDSCAPE	DIN-A4 landscape	297 x 210
*A3	DIN-A3	297 x 420
*A3-LANDSCAPE	DIN-A3 landscape	420 x 297
*A5	DIN-A5	148 x 210
*A5-LANDSCAPE	DIN-A5 landscape	210 x 148
*PARAMETERS(...)	User-specific	max. 2040 x 2040

PAGE-SIZE = *PARAMETERS(...)

Specifies the width and length of a PDF page explicitly.

WIDTH = <integer 2..2040>

Specifies the page width in mm.

HEIGHT = <integer 2..2040>

Specifies the page length in mm.

MARGINS = *PARAMETERS(...)

Defines the distances to the margins.

LEFT = 20 / <integer 0..2040>

Specifies the distance to the left-hand margin in mm.

RIGHT = 20 / <integer 0..2040>

Specifies the distance to the right-hand margin in mm. Data lines which extend beyond the right-hand margin are truncated or wrapped according to the specification in the LINE-TRUNCATION operand.

TOP = 20 / <integer 0..2040>

Specifies the distance to the top margin in mm.

BOTTOM = 20 / <integer 0..2040>

Specifies the distance to the bottom margin in mm.

DENSITY = *PARAMETERS(...)

Defines the line density.

LINES-PER-INCH = 6 / <integer 3..24>

Number of lines which are to be output per inch.

FONT = *PARAMETERS(...)

Determines the font which is to be used.

NAME = *COURIER / *HELVETICA / *TIMES

Specifies the name of the font.

CHARACTER-STYLE = *NORMAL / *BOLD / *ITALIC / *BOLD-ITALIC

Specifies the character style (normal, bold, italics, or bold and italics).

SIZE = g / <integer 1..72>

Specifies the font size in points (pt).

LINE-TRUNCATION = *YES / *NO

Specifies whether data lines which extend beyond the right-hand margin are truncated (see also definition of the right-hand margin in the MARGINS operand).

The default is *YES, i.e. longer data lines are truncated. When data lines are truncated, a message to this effect is issued after conversion.

LINE-TRUNCATION = *NO

Longer data lines are wrapped. The line break takes place at the word which extends beyond the margin. Thereby a word is a string which is limited by blanks, punctuation marks or the margin.

OVERLAY = *NONE / *PARAMETERS(...)

Specifies whether a background picture is to be used.

OVERLAY = *PARAMETERS(...)

The PDF pages are to contain a background picture.

FROM-FILE = <filename 1..54 without-gen-vers>

File which contains the background picture.



Only pictures in JPG format are supported. Transfer of the pictures from the PC to BS2000 must take place in binary format.

FRAME = *PAGE / *TEXT / *CUSTOM(...)

Determines the frame in which the background picture is positioned.

FRAME = *PAGE

The background picture is positioned within the physical page (determined by the specifications in the PAGE-SIZE operand).

FRAME = *TEXT

The background picture is positioned within the text frame (determined by the specifications in the MARGINS operand).

FRAME = *CUSTOM(...)

Defines a frame by means of the distances to the margins. This frame is independent of the text frame. However, the preset values are the same as the defaults in the PAGE-SIZE and MARGINS operands.

LEFT = 0 / <integer 0..2040>

Specifies the distance to the left-hand margin in mm.

RIGHT = 0 / <integer 0..2040>

Specifies the distance to the right-hand margin in mm.

TOP = 0 / <integer 0..2040>

Specifies the distance to the top margin in mm.

BOTTOM = 0 / <integer 0..2040>

Specifies the distance to the bottom margin in mm.

HORIZONTAL-ALIGNMENT = *LEFT / *RIGHT / *CENTER

Determines the horizontal alignment of the picture within the frame.

HORIZONTAL-ALIGNMENT = *LEFT

The picture is aligned with the left-hand margin of the frame.

HORIZONTAL-ALIGNMENT = *RIGHT

The picture is aligned with the right-hand margin of the frame.

HORIZONTAL-ALIGNMENT = *CENTER

The picture is centered horizontally in the frame.

VERTICAL-ALIGNMENT = *TOP / *BOTTOM / *CENTER

Determines the vertical alignment of the picture within the frame.

VERTICAL-ALIGNMENT = *TOP

The picture is aligned with the top.

VERTICAL-ALIGNMENT = *BOTTOM

The picture is aligned with the bottom.

VERTICAL-ALIGNMENT = *CENTER

The picture is centered vertically in the frame.

SCALE = *UNCHANGED

Determines the size of the picture.

The default is *UNCHANGED, i.e. the picture size is not changed.

SCALE = *FIT-HEIGHT

The picture is fitted to the height of the frame.

SCALE = *FIT-WIDTH

The picture is fitted to the width of the frame.

SCALE = *FIT-FRAME

The picture is fitted to the height and width of the frame. If the relationships of scale of the frame and picture differ, the picture may be distorted.

PRESENTATION = *SPOOL-PARAMETERS(...)

Specifies spool parameters which control the layout of the PDF file (analogously to print output with the PRINT-DOCUMENT command).

RECORD-PART = *ALL / *PARAMETERS(...)

Defines whether the records of the text file are to be processed in full or if only a particular part of each record is to be processed. When the default value *ALL is specified, the records are processed in full.

RECORD-PART = *PARAMETERS(...)

Specifies which part of the records is to be processed. Only the specified part is taken into account during conversion to PDF. This entry can be used, for example, to omit the ISAM key or control characters in the PDF file.

FIRST-CHARACTER = 1 / <integer 2..32767>

Allows a byte number (record column) to be specified indicating the point as of which the records of a file are to be output. The bytes of a record are numbered consecutively from left to right starting with 1; ISAM keys and control characters are components of a record.

LAST-CHARACTER = *STD / <integer 1..32767>

Specifies the byte indicating the point at which printing of each record is to stop.

LINE-PER-PAGE = *STD / <integer 1..32767>

Specifies how many lines (including blank lines) are to be printed on a page.

LINE-PER-PAGE = *STD

If the operand is omitted, the number of lines per print page is calculated using the following formula: Number of lines = P * L - N - 6

The name sections have the following meanings:

P = paper size in inches

L = line density

N = number of line before the first channel 1

If the value specified for the LINE-PER-PAGE operand is greater than the specified number of lines in the loop, the value in the loop is used.

LINE-SPACING = 1 / 2 / 3 / *BY-EBCDIC-CONTROL(...) / *BY-IBM-CONTROL(...) / *BY-ASA-CONTROL(...)

Specifies the number of line feeds and the way in which control characters are interpreted.

LINE-SPACING= 1 / 2 / 3

The records are to be printed out with 1-, 2- or 3-line spacing.

LINE-SPACING = *BY-EBCDIC-CONTROL(...) / *BY-IBM-CONTROL(...) / *BY-ASA-CONTROL(...)

The contents of the first byte of each record are to be interpreted as EBCDIC, IBM or ASA feed control characters (refer to PRINT-DOCUMENT command).

CONTROL-CHAR-POS = *STD / <integer 1..2040>

Number of the data byte which is interpreted as the feed control character. In the case of records of variable length, the fields containing the length information are not counted as data.

FORM-NAME = *STD / <alphanum-name 1..6>

Determines the page format of the PDF file by specifying the form. The spool parameter file must contain a standard form for the printer type HP-90. The default setting *STD causes the standard form STD to be used for the printer type specified in the PRINTER-TYPE operand.

FORM-NAME = <alphanum-name 1..6>

Name of the form. Depending on the ROTATION operand, either the normal loop or the loop for page rotation from the form definition is used.

A loop is named implicitly in the form specification. The assigned loop must be contained in the printer control file \$SYSSPOOL.PRFILE.

The loop named implicitly via the FORM-NAME operand is ignored if the LOOP-NAME operand is specified at the same time. When FORM-NAME=*STD and LOOP-NAME=*STD, the standard form entered for the specified printer is printed.

Forms are created with the SPSERVE utility routine. The SHOW-SPOOL-FORMS command displays information on forms.

LOOP-NAME = *STD / <alphanum-name 1..3>

Name of the loop to be loaded into the feed information buffer (VFB/FCB). The loop name must not include the characters '\$', '&' or '@'.

The default setting *STD causes the form's default loop to be used for the specified printer type.

Loops are stored in the PRFILE printer control file. They are created and managed using the PRM utility routine.

LOOP-NAME = <alphanum-name 1..3>

Name of the loop which is to control line feed. The length of the specified loop must match the length of the default loop of the form used.

A loop for LP- HP printers is selected from the PRFILE in accordance with the specification in the PRINTER-TYPE operand.

CHARACTER-SET = *STD / <alphanum-name 1..3>

Name of the font which is to be used for conversion. Courier is the font which is always used. Only the font properties WEIGHT and CHARACTER-STYLE are evaluated for PDF conversion. The table below shows the character style resulting from this (see the CHARACTER-STYLE operand in the direct parameters):

Font property WEIGHT=	Font property CHARACTER-STYLE=	
	*ITALICS	Other value
*BOLD	*BOLD-ITALICS	*BOLD
Other value	*ITALICS	*NORMAL

The default setting *STD causes the standard font to be used for the specified printer type. It can be displayed by means of SHOW-SPOOL-FORMS.

Fonts are created with the SPSERVE utility routine. The SHOW-SPOOL-CHARACTER-SETS displays information on the fonts.

PRINTER-TYPE = *HP90-PRINTER / *HP-PRINTER / *LP-PRINTER

Determines which form is to be used for PDF conversion via the printer type.

PRINTER-TYPE = *HP90-PRINTER

The form for HP90 printers is used.

PRINTER-TYPE = *HP-PRINTER

The form for HP printers is used.

PRINTER-TYPE = *LP-PRINTER

The form for LP printers is used.

LEFT-MARGIN = 20 / <integer 0..2040>

The output text is to be indented from the left margin by the specified number of millimeters. An indent of 20 millimeters is preset.

ROTATION = *NO / *YES

Specifies whether page rotation is to be executed.

ROTATION = *NO

Page rotation is not performed. Any control characters for page rotation in the file are not interpreted. The loop for conversion is determined as follows:

- When a loop is specified explicitly in the LOOP-NAME operand, the specified loop is used.
- If no loop was specified (corresponds to LOOP=*STD), the form definition is evaluated:
 - If a form was specified in the FORM operand, the loop defined for VERTICAL-CONTROL is used.
 - If no form was specified (corresponds to FORM=*STD), the standard form for the corresponding printer type (see the PRINTER-TYPE operand) is evaluated and the loop defined for VERTICAL-CONTROL is used.

ROTATION = *YES

Page rotation is executed. Any control characters for page rotation contained in the file are evaluated. The loop for conversion is determined as follows:

- When a loop is specified explicitly in the LOOP-NAME operand, the specified loop is used.
- If no loop was specified (corresponds to LOOP=*STD), the form definition is evaluated:
 - If a form was specified in the FORM operand, the loop defined for ROTATION-CONTROL is used.
 - If no form was specified (corresponds to FORM=*STD), the standard form for the corresponding printer type (see the PRINTER-TYPE operand) is evaluated and the loop defined for VERTICAL-CONTROL is used.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed without error
	1	CMD0202	Syntax or semantic error in command

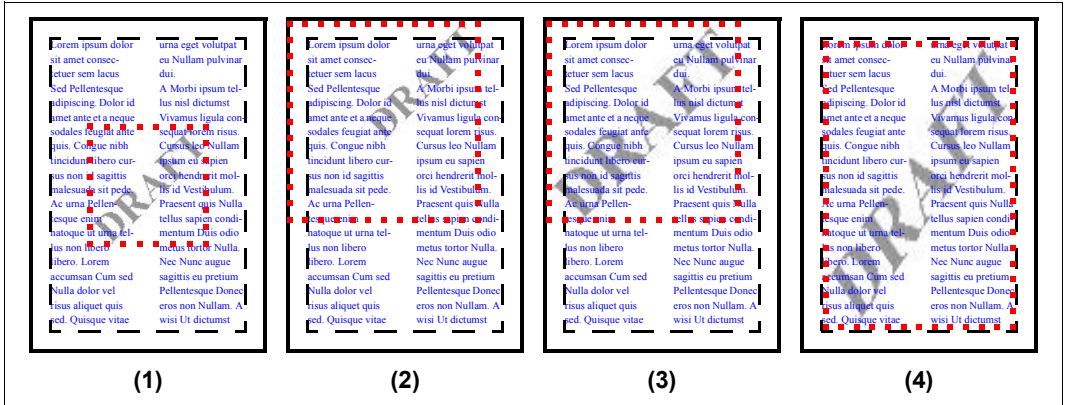
Note

The CCS of the specified file is used for conversion. If no file-specific CCS is defined, the CCS defined for the user ID is used. If there is no user-specific CCS, the defined system-global CCS is used.

Only those characters from the CCS are displayed which are also contained in the Windows Code Page WCP1252P (see the "XHCS" manual [51]).

Examples of how the OVERLAY operand is used

The figure below shows four examples of PDF pages on which the background picture draft.jpg is used:



- (1) Positions the background picture draft.jpg in its original size (70 x 70 mm) in the center of the page. Specification in the OVERLAY operand:

```
overlay=*par(from-file=draft.jpg, frame=*page,
horizontal-alignment=*center, vertical-alignment=*center)
```

- (2) Positions the background picture draft.jpg in its original size at the top right in a user-specific frame. The frame is square (130 x 130 mm) and begins at the top left of the DIN A4 page. Specification in the OVERLAY operand:

```
overlay=*par(from-file=draft.jpg,
frame=*custom(left=0,right=80,top=0,bottom=167),
horizontal-alignment=*right, vertical-alignment=*top)
```

- (3) Positions the background picture draft.jpg in the same user-specific frame as in example 2. Here the picture is fitted to the frame size. As the scale of the frame matches that of the picture, no distortion occurs. Specification in the OVERLAY operand:

```
overlay=*par(from-file=draft.jpg,
frame=*custom(left=0,right=80,top=0,bottom=167), SCALE=*FIT-FRAME)
```

- (4) Positions the background picture draft.jpg in the text frame and fits the picture to the frame size. In the case of a DIN A4 page with 20 mm spacing, the text frame is 170 mm wide and 257 mm high. As this scale does not match that of the picture, distortion occurs here. Specification in the OVERLAY operand:

```
overlay=*par(from-file=draft.jpg, frame=*text, SCALE=*FIT-FRAME)
```


COPY-FILE

Copy files

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	TSOS SECURITY-ADMINISTRATION SAT-FILE-MANAGEMENT SAT-FILE-EVALUATION STD-PROCESSING HARDWARE-MAINTENANCE
Routing code:	\$ (bei NBCONOPI=N) bzw. E (bei NBCONOPI=Y)

Function

The COPY-FILE command copies files, file generations or file generation groups one block at a time, without changing them. The command can therefore not be used to modify file attributes. For example, a SAM file cannot be converted to an ISAM file, nor can fixed length records be changed to variable length ones. It is, however, possible to change the BLOCK-CONTROL-INFO file attribute (see the BLOCK-CONTROL-INFO operand). Tape files with BLOCK-CONTROL-INFO=*PAMKEY must therefore have the default block format so that they can be processed using COPY-FILE.

If the target file (copy) is not yet cataloged, it is automatically created on a public volume when the COPY-FILE command is executed (as with a CREATE-FILE command with default values for the specified target file).

If the target file is to be stored on a different volume (Net-Storage volume, private disk or tape), it must be created using the CREATE-FILE command (DEVICE-TYPE, VOLUME operands) before the COPY-FILE command is issued.

The COPY-FILE command uses the internal file link names DMCOPY11 (FROM-FILE) and DMCOPY22 (TO-FILE). Both of these two link names are released again after processing, even if the user explicitly specified them. Thus if wildcards are used in the file name, any existing TFT entry is applicable only during copying of the first file to be processed.

If the target file is a disk file which has not yet been cataloged, the primary and secondary allocations are taken from the original disk file (see also the CREATE-FILE command, PRIMARY-ALLOCATION operand). If the original file is on tape, the target disk file will be given a default allocation.

The COPY-FILE command will be rejected if the target file is a read-only file (ACCESS=READ or no write authorization when access is controlled using BASIC-ACL or GUARDS), if a password has been specified to protect it against unauthorized write access, if a retention period (*EXPIR-DATE* > today's date) has been specified for it, or if the secondary allocation for the output file has a null value and the primary allocation is insufficient.

The user may specify whether the protection attributes of the original file are to be transferred to the target file (PROTECTION operand).

Privileged functions

A user working with the TSOS privilege can select the IGNORE-PROTECTION operand to explicitly circumvent any file protection on the source and target files which would prevent copying of the file.

Verifying command execution

The user can verify command execution. Use of the verification mechanisms is particularly helpful when a set of files is specified (using a partially qualified file name or wildcards).

The REPLACE-OLD-FILES operand governs whether a verification check is carried out before an existing target file is overwritten.

The DIALOG-CONTROL operand governs whether a verification dialog is to be conducted with the user. The OUTPUT operand lets the user specify whether each file name that is processed is to be logged in a guaranteed message to SYSOUT. The processed file names can be viewed in the message stream (see the ASSIGN-STREAM command, STREAM-NAME=SYSMMSG operand, or the EXECUTE-CMD command, MSG-STRUCTURE-OUTPUT operand, in the "SDF-P" manual [34]).

File generation groups

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

- a) The group entries for the two file generation groups match each other (i.e. the values for *MAXIMUM*, *FIRST-GEN*, *LAST-GEN* and *BASE-NUM* in the catalog entry are the same). In the file generation group into which DMS is to write the copy, the generations from *FIRST-GEN* to *LAST-GEN* must already be cataloged.
- b) The value for *MAXIMUM* is the same for both file generation groups, and the file generation group into which DMS is to write the copy contains no generations (i.e. the fields *FIRST-GEN*, *LAST-GEN* and *BASE-NUM* in the catalog entry have the value zero).

A file generation group which is to be copied may not contain any generations on tape.

A file generation group can only be copied into a single file or a file generation if it consists of SAM file generations which have the same attributes (e.g. same record and block lengths, same record format, same BLK-CONTR value). Furthermore, the target file generation must not be part of the file generation group which is to be copied.

Files held on private disks

If a file on a private disk has an entry in the system catalog, but no entry in the F1 label, the catalog entry is deleted when the file is copied. If this file is the input file, the command is rejected. If it is the target file, a new file is created in public space. Any COPY-FILE command for an ISAM file held on private disk with separate index and data areas will be rejected.

Tape files

- K tape files Tape files for which *BLK-CONTR=PAMKEY* must have the standard block format (catalog entry *BUF-LEN=STD(n)* if the COPY-FILE command is to be used on them.
- NK tape files Tape files for which *BLK-CONTR=DATA/NO* and for which the *BUF-LEN* value is a multiple of 2048 bytes can be copied.
- External files on tape When NK files are copied to tape, the *BLK-CONTR* information is removed if the catalog entry is deleted. If a tape file is to be copied which is not cataloged, it must first be cataloged by an IMPORT-FILE command. If the file is an NK file (*BLK-CONTR=DATA/NO*), the user must supply the correct BLOCK-CONTROL-INFO operand in the ADD-FILE-LINK command, i.e. he must specify BLOCK-CONTROL-INFO=*NO/*WITHIN-DATA-BLOCK to correspond to the actual file format, and must specify LINK-NAME = DMCOPY11 and the ACCESS-METHOD operand.

If a K file (*BLK-CONTR=PAMKEY*) is inadvertently copied as an NK file (*BLK-CONTR=DATA*), the resulting disk file will be unreadable; this is because the first 16 bytes on each PAM page, which for *BLK-CONTR=PAMKEY* would hold data, will have been overwritten with management information.

Internally, the COPY-FILE command uses the access method UPAM, which does not support the processing of continuation tapes. It is thus possible to copy several files to one tape. It is not possible however, to copy files which extend across multiple tapes.

PLAM libraries

PLAM libraries can only be copied if the library does not contain any specially protected data (see the “LMS” manual [21]).

A PLAM library protected by BASIC-ACL or GUARDS in a remote system cannot be copied.

Remote file access (see the “RFA” manual [31])

Copying from one remote system to another, with the input and output on different systems, is supported by a higher-level execution routine. The local system acts, in this case, only as an intermediate station for data transfer.

A SET-RFA-CONNECTION command must be issued for both of the remote systems before copying is started.

When a remote file is copied to a local one with the operand PROTECTION=*SAME, the passwords will not be transferred.

If different versions of BS2000, the following basic rules apply:

- The COPY-FILE should be issued on the system with the highest version.
- The file should be converted appropriately, either before or afterwards, using the PAMCONV utility.
- The specified operands must be supported by the operating system version of the remote system.
- If a file is to be copied from one remote system to another and I/O on two different systems is involved, the operation is supported by the higher-ranking execution routine. The local system serves as an intermediate system for the data transfer.
- The SET-RFA-CONNECTION command must be issued for both remote systems before copying starts.

File encryption

Normally it is not necessary to enter the crypto password in the task’s crypto password table in order to copy encrypted files as the file content is transferred without being decrypted.

Decryption during copying is required for the following copying operations:

- Encrypted files are copied to tape or private disk.
- Encrypted files are copied to a system < BS2000/OSD-BC V6.0 via RFA.
- Shared update processing was defined in the TFT entry of the internal file link name DMCOPY11 or DMCOPY22 (see the ADD-FILE-LINK command).

Format

COPY-FILE	Alias: CPF
<p>FROM-FILE = <filename 1..54 with-wild(80)></p> <p>,TO-FILE = <filename 1..54 with-wild-constr(80)></p> <p>,PROTECTION = <u>*STD</u> / *SAME</p> <p>,CHANGE-DATE = <u>*STD</u> / *SAME</p> <p>,REPLACE-OLD-FILES = <u>*YES</u> / *NO / *BY-DIALOG</p> <p>,BLOCK-CONTROL-INFO = <u>*KEEP-ATTRIBUTE</u> / *IGNORE-ATTRIBUTE / *CHECK-REMOVAL</p> <p>,IGNORE-PROTECTION = <u>*NO</u> / list-poss(2): *SOURCE-FILE / *TARGET-FILE</p> <p>,DIALOG-CONTROL = <u>STD</u> / *NO / *ERROR / *FILE-CHANGE / *MORE-THAN-ONE-FILE / *USER-ID-CHANGE / *CATALOG-CHANGE</p> <p>,OUTPUT = <u>*NO</u> / *SYSOUT</p>	

Operands

FROM-FILE = <filename 1..54 with-wild(80)>

Name of the file, file generation or file generation group which is to be copied (the original file). If the file name includes wildcards, or if the specification is partially qualified, the command is executed for the associated set of files. Processing can be controlled and verified with the aid of the DIALOG-CONTROL and OUTPUT operands. Only systems support is allowed to use wildcards within the user ID. If the original file (FROM-FILE) is not cataloged under the user ID for the job, the relevant user ID must be specified in the command, and the user ID under which the command is called must be authorized to access this file/file generation/FGG (USER-ACCESS= ALL-USERS, or read access rights using BASIC-ACL or GUARDS, or co-owner).

If the original file is a file generation group, the target file (TO-FILE) must also be a FGG (with the sole exception of a FGG (FROM-FILE) which consists of SAM file generations which all have the same attributes for record format, record and block length, and the block control information). In this case only, it is possible to copy into a single file or file generation. The target file generation may not be a constituent of the file generation group which is being copied.

TO-FILE = <filename 1..54 with-wild-constr(80)>

Name of the file, file generation or file generation group into which the copy is to be made (the target file). If a set of files has been specified in the FROM-FILE operand, a constructor string can be used to specify how the names of the target files are to be formed. The constructor string should be designed to ensure that the names of the target files are unique. If, for example, a fully qualified file name is specified as the target file for a set of files, the first source file will be copied to this file. For the following source files, the target

file will already exist. Depending on the setting of the REPLACE-OLD-FILES operand, either every subsequent source file will likewise be copied to the target file, or copying will be rejected. When processing of the command is complete, the target file will hold the contents either of the first source file or of the last source file.

If the target file has not yet been cataloged, then only the user's own ID (the one under which the LOGON command was issued) or a user ID for which the user is co-owner may be specified. If a target file which has not yet been cataloged is to be stored on an SM subset, the system attempts to locate it on a suitable volume set on the basis of the source file's attributes (performance, availability). If the target file is already cataloged under another user's ID, this user ID must be specified, and the user ID under which the command is called must be authorized to access this file (i.e. USER-ACCESS=ALL-USERS or the appropriate access rights using BASIC-ACL or GUARDS must have been assigned). If the target file is a file generation group, then the original file must also be a file generation group.

If the file is to be copied to a private volume, then the target file must have been appropriately cataloged beforehand (CREATE-FILE).

The COPY-FILE command will be rejected in the following cases:

- if the target file may only be read; i.e. if ACCESS=READ is defined or if write access is prohibited using BASIC-ACL or GUARDS;
- if there is still a retention period applicable to the file (*EXPIR-DATE* > current date);
- if the target file is password-protected against unauthorized write access;
- if the primary allocation for a target file on disk is insufficient and the secondary allocation is defined as 0.

PROTECTION = *STD / *SAME

Specifies whether the same attributes for file protection and file security are to be set up for the target file as for the original one.

The coding table entered in the catalog (*COD-CH-SET*) is always entered in the catalog entry of the target file.

The performance attributes of the source file are **not** transferred (see output fields *IO(PERF)*, *IO(USAGE)*, *DISK-WRITE*, SHOW-FILE-ATTRIBUTES command).

PROTECTION = *STD

Specifies that the attributes of the original file for file protection and file security should *not* be applied to the target file. If the target file is to be newly created, it will be given the default attributes arising from the CREATE-FILE command. If the target file already exists, it will retain its previous protection attributes.

PROTECTION = *SAME

The target file is assigned the same file security and protection attributes as the source file (i.e. for *BACKUP-CLASS*, *SAVED-PAG*, *EXPIR-DATE*, *EXPIR-TIME*, *DESTROY*, *FREE-FOR-DEL*, *MANAGE-CLASS*, *USER-ACC*, *ACCESS*, and the same passwords). The *AUDIT* value and the lock to prevent the release of memory space (*SP-REL-LOCK=YES*) are not transferred. For a *BASIC-ACL* and defined *GUARDS*, a distinction must be made between the following cases:

- Target file on public disk or Net-Storage volume:
The file protection attributes set with *BASIC-ACL* or *GUARDS* are transferred.
- Target file on private disk:
If access is controlled by a *GUARDS*, system default attributes are set (*USER-ACCESS=OWNER-ONLY*, *ACCESS=WRITE*). If the file is only protected by a *BASIC-ACL*, the *BASIC-ACL* is transferred.
- Target file on tape:
Protection attributes are always set to system default values (*USER-ACCESS=ALL-USERS*, *ACCESS=WRITE*).

The operand value *PROTECTION=*SAME* will be ignored in the following cases:

- The target file is a temporary file.
- The target file is a file generation (for which the attributes for protection and security are defined in the group entry).
- A nonprivileged user specifies a target file with a foreign user ID.

As management classes are pubset-specific they can only be taken over if the source and target files reside on the same pubset. When copying to another pubset an existing target file retains its management class; a new target file is assigned *MANAGEMENT-CLASS=*NONE* (as with *PROTECTION=*STD*).

If a temporary file is copied into a permanent one and *PROTECTION=*SAME* is specified, the permanent file will be given the attribute *BACKUP=E*, i.e. the new file will be ignored when *ARCHIVE* backups are carried out. The value for *BACKUP* must be amended by a *MODIFY-FILE-ATTRIBUTES* command if the file is to be backed up automatically during *ARCHIVE* runs.

The following applies for nonprivileged users when copying from a foreign user ID: If the original file is protected with *BASIC-ACL* or *GUARDS* (see the “*SECOS*” manual [35]), the target file’s protection attributes *USER-ACCESS*, *ACCESS*, *BASIC-ACL* and *GUARDS* are set to system default values (see the *CREATE-FILE* command).

CHANGE-DATE = *STD / *SAME

Specifies whether the target file is to be given the same change date (*CHANGE-DATE*) as the source file.

CHANGE-DATE = *STD

The change date of the target file will be updated.

CHANGE-DATE = *SAME

The change date of the source file will be transferred to the target file. The specification will apply in the following cases:

- The target file resides under a foreign user ID.
- The target file is a file generation.

REPLACE-OLD-FILES = *YES / *NO / *BY-DIALOG

The user can specify here whether, if there is already a file cataloged under the name of the target file, it should be overwritten or whether the system should ask at execution time whether to overwrite any such file (REPLACE-OLD-FILES=*BY-DIALOG).

In *batch mode* REPLACE-OLD-FILES=*NO always applies.

If the target file is a *tape file*, the operand REPLACE-OLD-FILES will be ignored. Any existing *tape file* with the same name will be overwritten with no message.

If the target file is *empty*, it will be overwritten without any message.

REPLACE-OLD-FILES = *YES

Specifies that any existing disk file should be overwritten without any message.

REPLACE-OLD-FILES = *NO

Any existing file will not be overwritten. The command is rejected. In procedures error handling is triggered (spin-off mechanism in non-S procedures or SDF-P error handling in S procedures).

REPLACE-OLD-FILES = *BY-DIALOG

The operand value *BY-DIALOG will only have any effect in interactive mode. It allows the user to decide in an interactive dialog with the system, whether to overwrite an existing target file or not. The query "OVERWRITE (Y/N?)" will be displayed. If the reply is "Y", the target file will be overwritten without any further message.

If the reply is "N", the target file is not overwritten and a message to this effect is issued.

BLOCK-CONTROL-INFO = *KEEP-ATTRIBUTE / *IGNORE-ATTRIBUTE / *CHECK-REMOVAL

Specifies whether there may be any difference in the file attribute BLOCK-CONTROL-INFO (*BLK-CONTR*) for the original and target files (or in the TFT entry DMCOPY22).

During copying, data is transferred in blocks, i.e. without regard to the internal block structure of the file. For this reason, a file in K format (*BLK-CONTR=PAMKEY*) should not be copied into a file in NK format (*BLK-CONTR=DATA* or *NO*), or vice versa. Such transfers should be carried out using a utility routine which takes into account the internal block structure of the file (e.g. PERCON, PAMCONV).

*KEEP-ATTRIBUTE is the default setting, i.e. both files (or the TFT entry for DMCOPY22) must have the same BLOCK-CONTROL-INFO attribute.

However, for files whose internal block structure does not depend on the BLOCK-CONTROL-INFO attribute, copying can be permitted (e.g. PLAM libraries). By specifying the operand values IGNORE-ATTRIBUTE or CHECK-REMOVAL, differing BLOCK-CONTROL-INFO attributes can be accepted.

It is recommended that `BLOCK-CONTROL-INFO=*CHECK-REMOVAL` is specified if the operand `BLOCK-CONTROL-INFO` has earlier been defined by an `/ADD-FILE-LINK LINK-NAME=DMCOPY22` command.

BLOCK-CONTROL-INFO = *KEEP-ATTRIBUTE

This attribute in the TFT entry for the target file, or in its catalog entry as appropriate, must match that for the original file. If this is not the case, the command is rejected.

BLOCK-CONTROL-INFO = *IGNORE-ATTRIBUTE

Allows any differences in the `BLOCK-CONTROL-INFO` attribute to be ignored in particular cases.

When a source file (`BLK-CONTR=PAMKEY`) is being copied to a file for which the `BLOCK-CONTROL` information is specified as `BLK-CONTR=NO` or `BLK-CONTR=DATA`, the PAM key will **not** be checked for possible user data.

This user data in the PAM key will no longer be available in the target file. If `BLK-CONTR=DATA` is specified for the target file, the first twelve bytes of each logical block which is copied will be overwritten with the `BLOCK-CONTROL` information.

Copying is possible in the following cases:

BLOCK-CONTROL-INFO in the output field BLK-CONTR (SHOW-FILE-ATTRIBUTES)	
Original file	Target file
PAMKEY	DATA (disk file only)
PAMKEY	NO
DATA (disk file only)	PAMKEY
NO	PAMKEY

BLOCK-CONTROL-INFO = *CHECK-REMOVAL

Any differences in the `BLOCK-CONTROL-INFO` attribute will only be ignored if no user data would be lost from the PAM key during copying. When an original file for which the `BLOCK-CONTROL-INFO` attribute is `BLK-CONTR=PAMKEY` is being copied into a target file in NK format (`BLK-CONTR=DATA` or `NO`), the PAM keys will be checked for user data. If the user part of the PAM key does contain data, then the `COPY-FILE` command will be rejected. This ensures that no user data will be lost.

Copying is possible in the following cases (if no user data is lost):

BLOCK-CONTROL-INFO in the output field BLK-CONTR (SHOW-FILE-ATTRIBUTES)	
Original file	Target file
PAMKEY	DATA (disk file only)
PAMKEY	NO

IGNORE-PROTECTION = *NO / *SOURCE-FILE / *TARGET-FILE

Specifies whether or not systems support wishes to circumvent the file protection, and if so for which file.

IGNORE-PROTECTION = *NO

The default option is that the existing file protection on the source and destination files will be observed.

IGNORE-PROTECTION = *SOURCE-FILE

The protection attributes of the source file, which the file's owner declared when creating the file, are to be ignored for privileged users (system administration). The protection may be by READ or EXEC-PASSWORD attributes, and by enhanced protection measures using BASIC-ACL or GUARDS.

IGNORE-PROTECTION = *TARGET-FILE

For privileged users (systems support), the protection attributes of the destination file are to be ignored when the source file is copied. The protection may be by ACCESS- or EXPIRATION-DATE, READ-/WRITE- or EXEC-PASSWORD attributes, and by enhanced protection measures using BASIC-ACL or GUARDS.

DIALOG-CONTROL = *STD / *NO / *ERROR / *FILE-CHANGE / *MORE-THAN-ONE-FILE / *CATALOG-CHANGE / *USER-ID-CHANGE

Specifies whether and under what conditions a verification dialog is to be conducted with the user during command execution. The verification dialog allows you to monitor the progress of the command and to take action if necessary. A control dialog is only possible in dialog mode but, in this mode, may also be used in procedures. In batch mode, all operand values have the same effect as *NO.

If one of the events (apart from *ERROR) specified in the DIALOG-CONTROL operand occurs, a verification dialog is initiated. Message DMS0810 is issued, asking you whether you want the command to be executed for the shown file or file set. The SDF abbreviation rules apply to the entered response. To display the possible responses, enter a question mark.

Syntax of replies to message DMS0810 (verification dialog)

```

mögliche Antworten: *YES(...) / *NO(...) / *TERMINATE

*YES(...)
  DIALOG-CONTROL = *UNCHANGED / *NO / *ERROR / *MORE-THAN-ONE-FILE /
                  *FILE-CHANGE / *CATALOG-CHANGE / *USER-ID-CHANGE
  ,OUTPUT = *UNCHANGED / *NO / *SYSOUT

*NO(...)
  DIALOG-CONTROL = *UNCHANGED / *NO / *ERROR / *MORE-THAN-ONE-FILE /
                  *FILE-CHANGE / *CATALOG-CHANGE / *USER-ID-CHANGE
  ,OUTPUT = *UNCHANGED / *NO / *SYSOUT

```

Meaning of the operands

The meaning of the DIALOG-CONTROL and OUTPUT suboperands is the same as that of the corresponding main command operands. The operand value *UNCHANGED leaves the previous setting unchanged.

Reply *YES(...)

The set of files listed in the message is processed. If you specify a new value in the DIALOG-CONTROL operand which calls for a reduction in the size of the set of files listed in the message (e.g. changing *CATALOG-CHANGE to *FILE-CHANGE), the file set is first slimmed down to reflect the new value, and then a second verification dialog is conducted for the reduced set of files.

The meaning of the DIALOG-CONTROL and OUTPUT suboperands is the same as that of the corresponding main command operands.

Reply *NO(...)

The set of files listed in the message is not processed. If you change the DIALOG-CONTROL operand, the verification check is not repeated until the next time the associated event occurs. If there are other files to process before the event occurs, they will be processed without verification.

Reply *TERMINATE

The set of files listed in the message is not processed and processing of the command terminates. Within a procedure, error handling (spin-off or SDF-P error handling) is initiated.

If an error occurs while one of the selected files is being processed and DIALOG-CONTROL has a setting other than *NO, an error verification dialog is initiated. Guaranteed message DMS0812 shows you the names of the source and target files in question and the DMS error code. You are then asked by message DMS0813 whether and if so how command execution is to continue. The SDF abbreviation rules apply to the entered response. To display the possible responses, enter a question mark.

Syntax of replies to message DMS0813 (error verification dialog)

```

mögliche Antworten: *YES(...) / *RETRY(...) / *NO / *SKIP-CATALOG(...) / *SKIP-USER-ID(...)

*YES(...)
  DIALOG-CONTROL = *UNCHANGED / *NO / *ERROR / *MORE-THAN-ONE-FILE /
                    *FILE-CHANGE / *CATALOG-CHANGE / *USER-ID-CHANGE
  ,OUTPUT = *UNCHANGED / *NO / *SYSOUT

*RETRY(...)
  DIALOG-CONTROL = *UNCHANGED / *NO / *ERROR / *MORE-THAN-ONE-FILE /
                    *FILE-CHANGE / *CATALOG-CHANGE / *USER-ID-CHANGE
  ,OUTPUT = *UNCHANGED / *NO / *SYSOUT

*SKIP-CATALOG(...)
  DIALOG-CONTROL = *UNCHANGED / *NO / *ERROR / *MORE-THAN-ONE-FILE /
                    *FILE-CHANGE / *CATALOG-CHANGE / *USER-ID-CHANGE
  ,OUTPUT = *UNCHANGED / *NO / *SYSOUT

*SKIP-USER-ID(...)
  DIALOG-CONTROL = *UNCHANGED / *NO / *ERROR / *MORE-THAN-ONE-FILE /
                    *FILE-CHANGE / *CATALOG-CHANGE / *USER-ID-CHANGE
  ,OUTPUT = *UNCHANGED / *NO / *SYSOUT

```

Meaning of the operands

The meaning of the DIALOG-CONTROL and OUTPUT suboperands is the same as that of the corresponding main command operands. The operand value *UNCHANGED leaves the previous setting unchanged.

Reply *YES(...)

Processing is to continue with the next file name.

Reply *RETRY(...)

Processing is to be retried for the same file name.

Reply *SKIP-CATALOG(...)

Processing is to continue with the file names from the next selected catalog. This selection only makes sense if wildcards were used in the catalog ID.

Reply *SKIP-USER-ID(...)

Processing is to continue with the file names from the next selected user ID. If file names of a user ID on different pubsets have been selected, processing is to continue with the file names from the next selected catalog ID. This selection only makes sense if wildcards were used in the user or catalog ID.

DIALOG-CONTROL = *STD

The default *STD setting is equivalent to *MORE-THAN-ONE-FILE in an interactive dialog (when SYSCMD is connected to the terminal) and to *NO in procedures and in batch mode.

DIALOG-CONTROL = *NO

All the selected files are processed without an opportunity for the user to intervene.

DIALOG-CONTROL = *MORE-THAN-ONE-FILE

A verification dialog is initiated if more than one file has been selected.

The verification query is issued for each of the selected files. If the catalog and/or user ID contains wildcards, the query is issued for each catalog and/or user ID.

DIALOG-CONTROL=*ERROR also applies implicitly.

DIALOG-CONTROL = *ERROR

An error verification dialog is initiated if an error occurs while one of the selected file names is being processed.

DIALOG-CONTROL = *FILE-CHANGE

A verification dialog is initiated for each selected file name. DIALOG-CONTROL=*ERROR also applies implicitly.

DIALOG-CONTROL = *CATALOG-CHANGE

The verification query is issued for all the selected file names in a catalog. DIALOG-CONTROL=*ERROR also applies implicitly.

DIALOG-CONTROL = *USER-ID-CHANGE

The verification query is issued for all the selected file names of a user ID within a catalog.

DIALOG-CONTROL=*ERROR also applies implicitly.

OUTPUT = *NO / *SYSOUT

Specifies whether the names of processed files are to be logged on SYSOUT.

OUTPUT = *NO

Only any errors that occur are logged on SYSOUT.

OUTPUT = *SYSOUT

The names of processed files and any errors that occur are logged on SYSOUT (guaranteed messages DMS0816 and DMS0812).

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed
2	0	DMS0546	Catalog entry for specified file has reached maximum size
2	0	DMS054A	Insufficient disk space or access to disk not possible
2	0	DMS05F5	Some blocks of the source file could not be copied
	1	CMD0202	Syntax or semantic error in command
	32	DMS0584	A state that does not allow the function to continue was reported during processing
	32	DMS05C7	Unexpected internal error in DMS

(Part 1 of 2)

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	64	DMS0585	Error detected when processing catalog or multiprocessor system Guaranteed message: DMS053C
	64	CMD0102	Interrupted by K2 key
	64	CMD0216	Privilege errors
	64	DMS0501	Requested catalog not available
	64	DMS0512	Requested catalog not found
	64	DMS051A	File already exists Guaranteed message: DMS051A
	64	DMS051B	Requested user ID not in pubset Guaranteed message: DMS051B
	64	DMS051C	User not authorized to access pubset Guaranteed message: DMS051C
	64	DMS0533	Requested file not cataloged in pubset Guaranteed message: DMS0533
	64	DMS0535	Specified file not shareable
	64	DMS057B	Invalid operand for migrated file
	64	DMS057C	Processing not possible due to HSMS error
	64	DMS057E	File has been migrated, and HSMS is not available
	64	DMS0586	It is not possible to access or reserve a volume at present
	64	DMS0587	Use of the specified command has been restricted by the system administrator
	64	DMS0588	The disk storage assignment could not be executed
	64	DMS05F8	DMS error reported Guaranteed message: DMS05F8
	64	DMS05F9	File attributes in source and target file not compatible
	64	DMS05FC	Specified user ID not in HOME pubset
	64	DMS0609	No access to system file
	64	DMS0610	Action with selection specification (wildcards): command execution failed for at least one of the selected files
	64	DMS0698	File generation groups do not have the same attributes
	64	DMS06B5	File already open or catalog entry not updated after system error
	64	DMS06B6	File attributes not compatible with file generation group
	64	DMS06C4	File generation group not yet cataloged
	64	DMS06D0	Specified file generation does not exist
	64	DMS06FF	BCAM connection severed
	128	DMS0506	Function not executed due to change in master
	130	DMS0524	System address space exhausted
	130	DMS053C	No space in pubset catalog file
	130	DMS0582	File is currently locked or being used and cannot be processed
	130	DMS0585	Error detected when processing catalog or multiprocessor system Guaranteed message: DMS053C
	130	DMS0586	It is not possible to access or reserve a volume at present
	130	DMS0588	The disk storage assignment could not be executed
	130	DMS0594	Not enough virtual memory available
	130	DMS05C8	Maximum permitted number of files reached

(Part 2 of 2)

Examples

*Example 1: Copying with PROTECTION=*SAME*

```

/sh-f-at paul.file.2,inf=(sec=*yes) ----- (1)
%000000003 :20SG:$USER1.PAUL.FILE.2
%----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% OWNER = R W X GROUP = - - - OTHERS = - - -
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-18
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 0 REL= 0 PAGES
/copy-file from=paul.file.2,to=karl.p-2 ----- (2)
/show-file-attr karl.p-2,sec=*yes ----- (3)
%000000003 :20SG:$USER1.KARL.P-2
%----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-19
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 0 REL= 0 PAGES
/copy-file from=paul.file.2,to=karl.p-2,prot=*same ----- (4)
/show-file-attr karl.p-2,sec=*yes ----- (5)
%000000003 :20SG:$USER1.KARL.P-2
%----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% OWNER = R W X GROUP = - - - OTHERS = - - -
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-19
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 0 REL= 0 PAGES
/show-file-attr proc.mini.2,inf=(sec=*yes) ----- (6)
%000000003 :20SG:$USER1.PROC.MINI.2
%----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = YES
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-18
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 0 REL= 0 PAGES
/show-file-attr bsp.proc.1,inf=(sec=*yes) ----- (7)
%----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = YES EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-18
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 0 REL= 0 PAGES
/copy-file from=proc.mini.2,to=bsp.proc.1,prot=*same ----- (8)
% DMS05F3 REQUIRED PASSWORD IS NOT IN PASSWORD TABLE. ENTER PASSWORD AND RETRY
COMMAND LATER
/add-pass 'maxi' ----- (9)
/copy-file from=proc.mini.2,to=bsp.proc.1,prot=*same ----- (10)
% DMS05F3 REQUIRED PASSWORD IS NOT IN PASSWORD TABLE. ENTER PASSWORD AND RETRY
COMMAND LATER
/add-pass 'otto' ----- (11)
/copy-file from=proc.mini.2,to=bsp.proc.1,prot=*same ----- (12)

```

```

/show-file-attr bsp.proc.1,inf=(sec=*yes) _____ (13)
%0000000003 :20SG:$USER1.BSP.PROC.1
%----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = YES
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-19
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 0 REL= 0 PAGES

```

- (1) Output of the file protection attributes of the file *PAUL.FILE.2*. The file has a BASIC-ACL entry. Access control is implemented via an BASIC-ACL (highest activated access protection).
- (2) Copies the file *PAUL.FILE.2* to the file *CARL.P-2*, thus creating its catalog entry with default attributes (see the CREATE-FILE command).
- (3) Output of the protection attributes for the file *CARL.P-2*. The file is only protected via the standard access control list (ACCESS and USER-ACCESS).
- (4) The file *PAUL.FILE.2* is copied to the file *CARL.P-2* again. In this case, PROTECTION=*SAME defines the transfer of protection attributes from the source file.
- (5) Output of the protection attributes for the file *CARL.P-2*. The file has the same protection attributes as the source file *PAUL.FILE.2* (see step 1).
- (6) Output of the protection attributes the file *PROC.MINI.2*. The file is protected by an execute password ('MAXI').
- (7) Output of the protection attributes the file *BSP.PROC.1*. The file is protected by a write password ('JOHN').
- (8) Attempts to copy the file *PROC.MINI.2* to the file *EXP.PROC.1*. The password for read access to *PROC.MINI.2* is not yet contained in the password table of the task. So the COPY-FILE command is rejected.
- (9) Entry of the password 'MAXI' into the password table of the task (see step 6).
- (10) Attempts to copy the file *PROC.MINI.2* to the file *EXP.PROC.1*. The password for write access to *EXP.PROC.1* is not yet contained in the password table of the task. So the COPY-FILE command is rejected.
- (11) Entry of the password 'JOHN' into the password table of the task (see step 7).
- (12) Attempts to copy the file *PROC.MINI.2* to the file *EXP.PROC.1*.
- (13) Output of the protection attributes the file *BSP.PROC.1*. Since PROTECTION=*SAME is specified, the file has the same protection attributes as the source file *PROC.MINI.2*, i.e. it is now protected by the execute password 'MAXI'.

Example 2: Copying a file generation group to a file

```

/show-file-attr max.group.2,select=(gen=*yes)
% 0 :20S2:$USER1.MAX.GROUP.2 (FGG)
% 3 :20S2:$USER1.MAX.GROUP.2(*0011)
% 3 :20S2:$USER1.MAX.GROUP.2(*0012)
% 3 :20S2:$USER1.MAX.GROUP.2(*0013)
%:20S2: PUBLIC: 4 FILES RES= 9 FRE= 6 REL= 0 PAGES
/show-file-attr max.group.2,inf=(org=*yes)
%0000000000 :20S2:$USER1.MAX.GROUP.2 (FGG)
% ----- GENERATION-INFO -----
% MAXIMUM = 3 BASE-NUM = 11 OVERFL-OPT = CYCL-REPL
% FIRST-GEN = 11 LAST-GEN = 13
%:20S2: PUBLIC: 1 FILE RES= 0 FRE= 0 REL= 0 PAGES
/show-file-attr max.file.1,inf=(org=*yes)
%0000000003 :20S2:$USER1.MAX.FILE.1
% ----- ORGANIZATION -----
% FILE-STRUC = NONE BUF-LEN = NONE BLK-CONTR = NONE
% IO(USAGE) = READ-WRITE IO(PERF) = STD DISK-WRITE = IMMEDIATE
% REC-FORM = NONE REC-SIZE = 0
% AVAIL = *STD
%:20S2: PUBLIC: 1 FILE RES= 3 FRE= 3 REL= 3 PAGES
/copy-file from=max.group.2,to=max.file.1
/show-file-attr max.file.1,inf=(org=*yes)
%0000000003 :20S2:$USER1.MAX.FILE.1
% ----- ORGANIZATION -----
% FILE-STRUC = SAM BUF-LEN = STD(1) BLK-CONTR = PAMKEY
% IO(USAGE) = READ-WRITE IO(PERF) = STD DISK-WRITE = IMMEDIATE
% REC-FORM = (V,N) REC-SIZE = 0
%:20S2: PUBLIC: 1 FILE RES= 3 FRE= 0 REL= 0 PAGES
/show-file max.file.1
*****
***** Inhalt von MAX.GROUP.2(*0011) *****
*****
***** (Ende)...
*****
***** Inhalt von MAX.GROUP.2(*0012) *****
*****
***** (Ende)...
*****
***** Inhalt von MAX.GROUP.2(*0013) *****
*****
***** (Ende)...
*****
% SH00301 WARNING: END OF FILE REACHED
e S*S0F+ 1( 1)

```

- (1) Output of the catalog entries for the file generation group *MAX.GROUP.2* with the associated generations. The existing generations are *0011, *0012, and *0013.
- (2) Output of the group entry with the specific attributes of a file generation group.
- (3) Output of the catalog entry for the file *MAX.FILE.1* with the file organization information. The file has not yet been opened for writing (as is evident from FILE-STRUC=NONE, for example).
- (4) Copies the file generation group *MAX.GROUP.2* to the file *MAX.FILE.1*.

- (5) Output of the catalog entry for the file *MAX.FILE.1* with the file organization information. The file was created as a SAM file and now occupies three PAM pages (see *FRE=0*).
- (6) Output of the content of file *MAX.FILE.1* using the command *SHOW-FILE*. The output is terminated with *E* (END).

Example 3: Copying a file generation group to a file generation group

```

/show-file-attr max.group.2,inf=(org=*yes) _____ (1)
%0000000000 :20S2:$USER1.MAX.GROUP.2 (FGG)
%----- GENERATION-INFO -----
% MAXIMUM = 3 BASE-NUM = 11 OVERFL-OPT = CYCL-REPL
% FIRST-GEN = 11 LAST-GEN = 13
%:20S2: PUBLIC: 1 FILE RES= 0 FRE= 0 REL= 0 PAGES
/cre-file-group group=max.group.2.copy,gen-par=(max=3) _____ (2)
/show-file-attr max.group.2.copy,inf=(org=*yes) _____ (3)
%0000000000 :20S2:$USER1.MAX.GROUP.2.COPY (FGG)
%----- GENERATION-INFO -----
% MAXIMUM = 3 BASE-NUM = 0 OVERFL-OPT = CYCL-REPL
% FIRST-GEN = 0 LAST-GEN = 0
%:20S2: PUBLIC: 1 FILE RES= 0 FRE= 0 REL= 0 PAGES
/copy-file from=max.group.2,to=max.group.2.copy _____ (4)
/show-file-attr max.group.2.copy,gen=*yes _____ (5)
% 0 :20S2:$USER1.MAX.GROUP.2.COPY (FGG)
% 3 :20S2:$USER1.MAX.GROUP.2.COPY(*0001)
% 3 :20S2:$USER1.MAX.GROUP.2.COPY(*0002)
% 3 :20S2:$USER1.MAX.GROUP.2.COPY(*0003)
%:20S2: PUBLIC: 4 FILES RES= 9 FRE= 6 REL= 0 PAGES
/show-file max.group.2.copy(*1) _____ (6)
*****
***** Inhalt von MAX.GROUP.2(*0011) *****
*****
.....(Ende)...
.....
% SH00301 WARNING: END OF FILE REACHED
e S*SOF+ 1( 1)
/show-file max.group.2.copy(*2) _____ (7)
*****
***** Inhalt von MAX.GROUP.2(*0012) *****
*****
.....(Ende)...
.....
% SH00301 WARNING: END OF FILE REACHED
e S*SOF+ 1( 1)
/show-file max.group.2.copy(*3) _____ (8)
*****
***** Inhalt von MAX.GROUP.2(*0013) *****
*****
.....(Ende)...
.....
% SH00301 WARNING: END OF FILE REACHED
e S*SOF+ 1( 1)

```

- (1) Output of the group entry *MAX.GROUP.2* with the specific attributes of a file generation group. The three existing generations are *0011, *0012, and *0013 (see LAST-GEN=13 or example 2).
- (2) Creates a group entry for the file generation group *MAX.GROUP.2.COPY*. The maximum number of generations is set to 3, as for *MAX.GROUP.2*.
- (3) Output of the newly created group entry *MAX.GROUP.2.COPY* with the specific attributes of a file generation group.
- (4) Copies the file generation group *MAX.GROUP.2* to the file generation group *MAX.GROUP.COPY*.
- (5) Output of the catalog entries for the file generation group *MAX.GROUP.2.COPY* with the associated generations. Generations *0001, *0002 and *0003 have been created.
- (6) Output of the file contents of the generation *MAX.GROUP.2.COPY(*0001)* with the SHOW-FILE command. The contents of the first generation of *MAX.GROUP.2*, i.e. the generation *0011, has been copied into the first generation. The output is terminated with *E* (END).
- (7) Output of the file contents of the generation *MAX.GROUP.2.COPY(*0002)* with the SHOW-FILE command. The contents of the second generation of *MAX.GROUP.2*, i.e. generation *0002, has been copied into the second generation. The output is terminated with *E* (END).
- (8) Output of the file contents of the generation *MAX.GROUP.2.COPY(*0003)* with the SHOW-FILE command. The contents of the third generation of *MAX.GROUP.2*, i.e. generation *0013, has been copied into the third generation. The output is terminated with *E* (END).

Example 4: Copying temporary files

```

/show-file-attr proc.mini.2,inf=(sec=*yes,backup=*yes) _____ (1)
%0000000003 :20SG:$USER1.PROC.MINI.2
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = YES
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-19
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE
% ----- BACKUP -----
% BACK-CLASS = A SAVED-PAG = COMPL-FILE VERSION = 1
% MIGRATE = ALLOWED
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 0 REL= 0 PAGES

/add-pass 'maxi' _____ (2)
/copy-file from=proc.mini.2,to=#temp.file.1,prot=*same _____ (3)
/show-file-attr #temp.file.1,inf=(sec=*yes,backup=*yes) _____ (4)
%0000000003 :20SG:$USER1.S.163.34JG.TEMP.FILE.1
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-19
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE
% ----- BACKUP -----
% BACK-CLASS = E SAVED-PAG = COMPL-FILE VERSION = 1
% MIGRATE = INHIBITED
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 0 REL= 0 PAGES
/copy-file from=#temp.file.1,to=max.file.2,prot=*same _____ (5)
/show-file-attr max.file.2,inf=(sec=*yes,backup=*yes) _____ (6)
%0000000003 :20SG:$USER1.MAX.FILE.2
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-19
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE
% ----- BACKUP -----
% BACK-CLASS = E SAVED-PAG = COMPL-FILE VERSION = 1
% MIGRATE = INHIBITED
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 0 REL= 0 PAGES
/show-file-attr max.file.3,inf=(sec=*yes,backup=*yes) _____ (7)
%0000000003 :20SG:$USER1.MAX.FILE.3
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% OWNER = R W X GROUP = - - - OTHERS = - - -
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = NONE
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = NONE
% SP-REL-LOCK= NO ENCRYPTION = *NONE
% ----- BACKUP -----
% BACK-CLASS = A SAVED-PAG = COMPL-FILE VERSION = 0
% MIGRATE = ALLOWED
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 3 REL= 3 PAGES
/copy-file from=#temp.file.1,to=max.file.3 _____ (8)
/show-file-attr max.file.3,inf=(sec=*yes,backup=*yes) _____ (9)
%0000000003 :20SG:$USER1.MAX.FILE.3
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% OWNER = R W X GROUP = - - - OTHERS = - - -
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-19
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE

```

```

% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG = COMPL-FILE  VERSION  = 1
% MIGRATE      = ALLOWED
%:20SG: PUBLIC:      1 FILE RES=          3 FRE=          0 REL=          0 PAGES

```

- (1) Output of the catalog entry for the file *PROC.MINI.2* with the attributes for file protection and file security. The file is accessible to all users (USER-ACCESS=ALL-USER) who also know the defined execute password 'MAXI'.
- (2) Entry of the password 'MAXI' into the password table of the task.
- (3) Copies the file *PROC.MINI.2* to the temporary file *TEMP.FILE.1* with PROTECTION=*SAME.
- (4) Output of the catalog entry for the temporary file *TEMP.FILE.1* with the file protection and backup attributes for the file. Default attributes have been set for the temporary file, as it is not possible to transfer a password, BACKUP-CLASS=A, or USER-ACCESS=ALL-USERS, e.g. (see also the CREATE-FILE command).
- (5) Copies the temporary file *TEMP.FILE.1* to the file *MAX.FILE.2* with PROTECTION=*SAME.
- (6) The catalog entry indicates that the file protection and backup attributes of the temporary file have been copied.
- (7) Output of the catalog entry for the file *MAX.FILE.3* with the file protection and backup attributes. The file is protected by means of a BASIC-ACL.
- (8) Copies the temporary file *TEMP.FILE.1* to the file *MAX.FILE.3*.
- (9) Output of the catalog entry for the file *MAX.FILE.3* with the file protection and backup attributes. These attributes have not been changed.

Example 5: Copying an existing file interactively

```

/copy-file from=max.file.1,to=max.file.10,replace-old=*by-dialog _____ (1)
% DMS0518 FILE ':20S2:$USER1.MAX.FILE.10' ALREADY EXISTS. OVERWRITE? REPLY (Y
=YES; N=NO)?n _____ (2)
% DMS0519 COPY COMMAND WITHDRAWN BY CALLER
/copy-file from=max.file.1,to=max.file.11,replace-old=*by-dialog _____ (3)

```

- (1) Specifies that the file *MAX.FILE.1* is to be copied to the file *MAX.FILE.10* and that a prompt is to be issued before overwriting any existing target file.
- (2) The file *MAX.FILE.10* already exists. *N* is entered in response to the prompt. So the COPY-FILE command is not executed.
- (3) Specifies that the file *MAX.FILE.1* is to be copied to the file *MAX.FILE.11* and that a prompt is to be issued before overwriting any existing target file. Since the target file does not exist in this case, the COPY-FILE command is executed without a prompt.

Example 6: Copying a file on private disk

```

/cre-file max.priv-file.1,sup=priv(vol=work01,dev-type=d3435) _____ (1)
/show-file-attr max.priv-file.1,inf=(org=*yes,alloc=*yes) _____ (2)
%0000000003*:20SG:$USER1.MAX.PRIV-FILE.1
% ----- ORGANIZATION -----
% FILE-STRUC = NONE          BUF-LEN   = NONE          BLK-CONTR = NONE
% IO(USAGE)  = READ-WRITE   IO(PERF)  = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = NONE          REC-SIZE  = 0
% AVAIL      = *STD
% ----- ALLOCATION -----
% SUPPORT    = PVT          S-ALLOC   = 9          HIGH-US-PA = 0
% EXTENTS    VOLUME        DEVICE-TYPE  EXTENTS    VOLUME    DEVICE-TYPE
%           1             WORK01      D3435
% NUM-OF-EXT = 1
%:20SG: PRDISC:          1 FILE RES=          3 FRE=          3 REL=          0 PAGES
/copy-file from=max.file.1,to=max.priv-file.1 _____ (3)
/show-file-attr max.priv-file.1,inf=(org=*yes,alloc=*yes) _____ (4)
% ----- ORGANIZATION -----
% FILE-STRUC = ISAM          BUF-LEN   = STD(1)        BLK-CONTR = DATA (2K)
% IO(USAGE)  = READ-WRITE   IO(PERF)  = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)        REC-SIZE  = 0
% KEY-LEN    = 8            KEY-POS   = 5
% AVAIL      = *STD
% ----- ALLOCATION -----
% SUPPORT    = PVT          S-ALLOC   = 2048       HIGH-US-PA = 3
% EXTENTS    VOLUME        DEVICE-TYPE  EXTENTS    VOLUME    DEVICE-TYPE
%           1             WORK01      D3435
% NUM-OF-EXT = 1
%:20SG: PRDISC:          1 FILE RES=          3 FRE=          0 REL=          0 PAGES

```

- (1) Creates a catalog entry for the file *MAX.PRIV-FILE.1*, which is to be stored on the private disk *WORK01*.
- (2) Output of the newly created catalog entry showing information on the file organization and the volume allocation. Since the file has not yet been opened, information on the file structure is not yet available (e.g. FILE-STRUC=NONE).
- (3) Copies the file *MAX.FILE.1* to the file *MAX.PRIV-FILE.1*.
- (4) Output of the catalog entry for the file *MAX.PRIV-FILE.1* showing information on the file organization and volume allocation. The file attributes (e.g. FILE-STRUC=ISAM) were entered during on executing the COPY-FILE command.

Example 7: Copying a file to a tape file (tape cartridge)

```

/show-file-attr max.tape-file.1,inf=(org=*yes,alloc=*yes) _____ (1)
%      :20S2:$USER1.MAX.TAPE-FILE.1
%      ----- ORGANIZATION -----
% FILE-STRUC = NONE      BUF-LEN = NONE      BLK-CONTR = NONE
% REC-FORM   = NONE      REC-SIZE  = 0
% CODE       = NONE      LABEL      = NONE      FILE-SEQ  = NONE
% BLK-OFFSET = 0
%      ----- ALLOCATION -----
% SUPPORT    = PVT
% EXTENTS    VOLUME      DEVICE-TYPE  EXTENTS    VOLUME      DEVICE-TYPE
% (          D2315K      TAPE-C4 )
%:20S2: TAPE : 1 FILE
/add-file-link link=dmcopy22,file-name=max.tape-file.1, _____ (2)
              sup=*tape(vol=*catalog)
/show-file-attr max.file.<4,5>,inf=(sec=*yes,org=y*es) _____ (3)
%0000000018 :20S2:$USER1.MAX.FILE.4
%      ----- SECURITY -----
% READ-PASS  = NONE      WRITE-PASS = NONE      EXEC-PASS = NONE
% USER-ACC   = OWNER-ONLY ACCESS      = WRITE      ACL        = NO
% OWNER      = R W X     GROUP        = - - -     OTHERS    = - - -
% AUDIT      = NONE      FREE-DEL-D = *NONE     EXPIR-DATE = 2012-03-19
% DESTROY    = NO        FREE-DEL-T = *NONE     EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO        ENCRYPTION = *NONE
%      ----- ORGANIZATION -----
% FILE-STRUC = SAM      BUF-LEN   = STD(1)     BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE IO(PERF)  = STD      DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)    REC-SIZE  = 0
%000000006 :20S2:$USER1.MAX.FILE.5
%      ----- SECURITY -----
% READ-PASS  = NONE      WRITE-PASS = NONE      EXEC-PASS = NONE
% USER-ACC   = OWNER-ONLY ACCESS      = WRITE      ACL        = NO
% OWNER      = R W X     GROUP        = - - -     OTHERS    = - - -
% AUDIT      = NONE      FREE-DEL-D = *NONE     EXPIR-DATE = 2012-03-19
% DESTROY    = NO        FREE-DEL-T = *NONE     EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO        ENCRYPTION = *NONE
%      ----- ORGANIZATION -----
% FILE-STRUC = SAM      BUF-LEN   = STD(1)     BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE IO(PERF)  = STD      DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)    REC-SIZE  = 0
%:20S2: PUBLIC: 2 FILES RES= 24 FRE= 5 REL= 3 PAGES
/show-file-link link=dmcopy22,inf=(file-contr=*yes) _____ (4)
%-- LINK-NAME ----- FILE-NAME -----
% T DMCOPY22 :20S2:$USER1.MAX.TAPE-FILE.1
%      ----- FILE-CONTROL-BLOCK - GENERAL -----
% ACC-METH = *BY-PROG OPEN-MODE = *BY-PROG REC-FORM = *BY-PROG
% REC-SIZE = *BY-PROG BUF-LEN = *BY-PROG BLK-CONTR = *BY-PROG
% F-CL-MSG = STD      CLOSE-MODE = *BY-PROG
%      ----- FILE-CONTROL-BLOCK - DISK -----
% SHARED-UPD = *BY-PROG WR-CHECK = *BY-PROG IO(PERF) = *BY-PROG
% IO(USAGE)  = *BY-PROG LOCK-ENV = *BY-PROG
%      ----- FILE-CONTROL-BLOCK - TAPE -----
% LABEL      = *BY-PROG (DIN-R-NUM = *BY-PROG, TAPE-MARK = *BY-PROG)
% CODE       = *BY-PROG EBCDIC-TR = *BY-PROG F-SEQ = *BY-PROG
% CP-AT-BLIM = *BY-PROG CP-AT-FEOV = *BY-PROG BLOCK-LIM = *BY-PROG
% REST-USAGE = *BY-PROG BLOCK-OFF = *BY-PROG TAPE-WRITE = *BY-PROG
% STREAM     = *BY-PROG
%      ----- FILE-CONTROL-BLOCK - ISAM -----
% KEY-POS    = *BY-PROG KEY-LEN = *BY-PROG POOL-LINK = *BY-PROG
% LOGIC-FLAG = *BY-PROG VAL-FLAG = *BY-PROG PROPA-VAL = *BY-PROG
% DUP-KEY    = *BY-PROG PAD-FACT = *BY-PROG READ-I-ADV = *BY-PROG
% WR-IMMED   = *BY-PROG POOL-SIZE = *BY-PROG
/copy-file from=max.file.5,to=max.tape-file.1,prot=*same _____ (5)

```

```

/show-file-attr max.tape-file.1,inf=(org=*yes,alloc=yes,sec=*yes) ----- (6)
%
% :20S2:$USER1.MAX.TAPE-FILE.1
% ----- HISTORY -----
% CRE-DATE = 2012-03-19 ACC-DATE = 2012-03-19 CHANG-DATE = 2012-03-19
% CRE-TIME = 00:00:00 ACC-TIME = 00:00:00 CHANG-TIME = 00:00:00
% ACC-COUNT = 1 S-ALLO-NUM = 0
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = ALL-USERS ACCESS = WRITE ACL = NO
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-19
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% ----- BACKUP -----
% BACK-CLASS = A SAVED-PAG = COMPL-FILE VERSION = 1
%
% ----- ORGANIZATION -----
% FILE-STRUC = SAM BUF-LEN = STD(1) BLK-CONTR = PAMKEY
% REC-FORM = (V,N) REC-SIZE = 2044
% CODE = EBCDIC LABEL = (STD,1) FILE-SEQ = 1
% BLK-OFFSET = 4
% ----- ALLOCATION -----
% SUPPORT = PVT BLK-COUNT = 5
% EXTENTS VOLUME DEVICE-TYPE EXTENTS VOLUME DEVICE-TYPE
% D2315K TAPE-C4
%:20S2: TAPE : 1 FILE
/cre-file max.tape-file.2,sup=*tape(vol=d2315k,dev-type=tape-c4) ----- (7)
/add-file-link link=dmcopy22,file-name=max.tape-file.2, ----- (8)
sup=tape(vol=*catalog,file-sequence=*new)
/show-file-link link=dmcopy22,inf=(file-contr=*yes) ----- (9)
%
% -- LINK-NAME ----- FILE-NAME -----
% T DMCOPY22 :20S2:$USER1.MAX.TAPE-FILE.2
% ----- FILE-CONTROL-BLOCK - GENERAL ATTRIBUTES -----
% ACC-METH = *BY-PROG OPEN-MODE = *BY-PROG REC-FORM = *BY-PROG
% REC-SIZE = *BY-PROG BUF-LEN = *BY-PROG BLK-CONTR = *BY-PROG
% F-CL-MSG = STD CLOSE-MODE = *BY-PROG
% ----- FILE-CONTROL-BLOCK - DISK FILE ATTRIBUTES -----
% SHARED-UPD = *BY-PROG WR-CHECK = *BY-PROG IO(PERF) = *BY-PROG
% IO(USAGE) = *BY-PROG LOCK-ENV = *BY-PROG
% ----- FILE-CONTROL-BLOCK - TAPE FILE ATTRIBUTES -----
% LABEL = *BY-PROG (DIN-R-NUM = *BY-PROG, TAPE-MARK = *BY-PROG)
% CODE = *BY-PROG EBCDIC-TR = *BY-PROG F-SEQ = NEW
% CP-AT-BLIM = *BY-PROG CP-AT-FEOV = *BY-PROG BLOCK-LIM = *BY-PROG
% REST-USAGE = *BY-PROG BLOCK-OFF = *BY-PROG TAPE-WRITE = *BY-PROG
% STREAM = *BY-PROG
% ----- FILE-CONTROL-BLOCK - ISAM FILE ATTRIBUTES -----
% KEY-POS = *BY-PROG KEY-LEN = *BY-PROG POOL-LINK = *BY-PROG
% LOGIC-FLAG = *BY-PROG VAL-FLAG = *BY-PROG PROPA-VAL = *BY-PROG
% DUP-KEY = *BY-PROG PAD-FACT = *BY-PROG READ-I-ADV = *BY-PROG
% WR-IMMED = *BY-PROG POOL-SIZE = *BY-PROG
/copy-file from=max.file.4,to=max.tape-file.2,prot=*same ----- (10)
/show-file-attr max.tape-file.2,inf=(org=*yes,alloc=*yes,prot=*yes) ----- (11)
%
% :20S2:$USER1.MAX.TAPE-FILE.2
% ----- HISTORY -----
% CRE-DATE = 2012-03-19 ACC-DATE = 2012-03-19 CHANG-DATE = 2012-03-19
% CRE-TIME = 00:00:00 ACC-TIME = 00:00:00 CHANG-TIME = 00:00:00
% ACC-COUNT = 1 S-ALLO-NUM = 0
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = ALL-USERS ACCESS = WRITE ACL = NO
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-19
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% ----- BACKUP -----
% BACK-CLASS = A SAVED-PAG = COMPL-FILE VERSION = 1
%

```



```

% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN   = STD(1)      BLK-CONTR = PAMKEY
% REC-FORM   = (V,N)       REC-SIZE  = 2044
% CODE      = EBCDIC       LABEL      = (STD,1)      FILE-SEQ  = 2
% BLK-OFFSET = 4
% ----- ALLOCATION -----
% SUPPORT    = PVT          VOLUME     DEVICE-TYPE EXTENTS   BLK-COUNT = 14
% EXTENTS    =             D2315K     TAPE-C4   EXTENTS   VOLUME   DEVICE-TYPE
%
%:20S2: TAPE :          1 FILE

```

- (1) Output of the catalog entry for the file *MAX.TAPE-FILE.1* showing information on the file organization and volume allocation. Only the catalog entry exists at this stage. The file has not yet been opened (e.g. FILE-STRUC=NONE). The data contained in the file is to be stored on the magnetic tape cartridge *D2315K* of volume type *TAPE-C4*. Since no volume has been reserved as yet, the volume ID is shown in parentheses.
- (2) Creates a TFT entry with the link name *DMCOPY22* for the file *MAX.TAPE-FILE.1*. The default link name of the COPY-FILE command is used for the target file.
- (3) Output of the catalog entries for the files *MAX.FILE.4* and *MAX.FILE.5* showing the file protection information. Both files have a BASIC-ACL entry as the highest activated access control. The USER-ACCESS attribute is set to all users in each case.
- (4) Output of the TFT entry with the link name *DMCOPY.22* indicating the file processing attributes.
- (5) Copies the file *MAX.FILE.5* to the tape file *MAX.TAPE-FILE.1* with PROTECTION=*SAME.
- (6) Output of the catalog entry for the file *MAX.TAPE-FILE.1* showing information on the file protection, file organization and volume allocation. The file attributes have been updated (e.g. FILE-STRUC=SAM). The file is located on volume *D2315K*. The BASIC-ACL could not be copied. The file is only protected by standard access control, with the USER-ACCESS attribute set to OWNER-ONLY.
- (7) Creates a catalog entry for the tape file *MAX.TAPE-FILE.2*, which is also to be stored on the tape cartridge *D2315K*.
- (8) Creates a TFT entry with the link name *DMCOPY22* for the file *MAX.TAPE-FILE.2*. The default link name of the COPY-FILE command is used for the target file. Since this is the second file to be stored on the tape cartridge, FILE-SEQUENCE=*NEW (or alternatively, FILE-SEQUENCE=2) must also be specified. Without this specification, the default value FILE-SEQUENCE=1 would be assumed when executing the copy operation, and the first file would be overwritten.
- (9) Output of the TFT entry with the link name *DMCOPY22* showing the relevant file processing information (see also the *F-SEQ* output field).

- (10) Copies the file *MAX.FILE.5* to the tape file *MAX.TAPE-FILE.2* with `PROTECTION=*SAME`.
- (11) Output of the catalog entry for the file *MAX.TAPE-FILE.2* showing the protection attributes, organization, and volume allocation for the file. The file attributes have been updated (e.g. `FILE-STRUC=SAM`). The file is located as the second file on volume *D2315K*. The `BASIC-ACL` could not be copied. The file is only protected by standard access control, with the `USER-ACCESS` attribute set to `*ALL-USERS`.

Example 8: Changing the BLOCK-CONTROL-INFO when copying

```

/show-file-attr ah.lib,org=*yes _____ (1)
%0000000273 :20SG:$USER1.ASS.PLAMLIB
% ----- ORGANIZATION -----
% FILE-STRUC = PAM          BUF-LEN   = STD(1)      BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE IO(PERF)  = STD        DISK-WRITE = IMMEDIATE
% TYPE       = PLAM-LIB
% AVAIL      = *STD
% WORK-FILE  = *NO          F-PREFORM = *NONE      SO-MIGR   = *ALLOWED
%:20SG: PUBLIC: 1 FILE RES= 273 FRE= 47 REL= 45 PAGES
/add-file-link link=dmcopy22,file-name=max.file.12,
               block-contr=*no _____ (2)
/copy-file from=ah.lib,to=max.file.12,block-contr=*check-removal _____ (3)
/show-file-attr max.file.12,org=*yes _____ (4)
%0000000273 :20SG:$USER1.MAX.FILE.12
% ----- ORGANIZATION -----
% FILE-STRUC = PAM          BUF-LEN   = STD(1)      BLK-CONTR = NO
% IO(USAGE)  = READ-WRITE IO(PERF)  = STD        DISK-WRITE = IMMEDIATE
% TYPE       = PLAM-LIB
% AVAIL      = *STD
% WORK-FILE  = *NO          F-PREFORM = *NONE      SO-MIGR   = *ALLOWED
%:20SG: PUBLIC: 1 FILE RES= 273 FRE= 47 REL= 45 PAGES

```

- (1) Output of the catalog entry for the file *ASS.PLAMLIB* showing the file organization. The file is a PLAM library that was created with `BLOCK-CONTROL-INFO=*PAMKEY`. The file is to be copied to an NK file with `BLOCK-CONTROL-INFO=*NO`. Since the internal block structure of the file does not depend on the `PAMKEY`, a different `BLOCK-CONTROL-INFO` is permitted when copying.
- (2) Creates a TFT entry with the link name *DMCOPY22* for the target file *MAX.FILE.12* with `BLOCK-CONTROL-INFO=*NO`. The default link name of the `COPY-FILE` command is used for the target file.
- (3) Copies the file *ASS.PLAMLIB* to the file *MAX.FILE.12* with `BLOCK-CONTROL-INFO=*CHECK-REMOVAL` (copy only if no user information from the `PAMKEY` is lost).
- (4) Output of the catalog entry for the target file *MAX.FILE.12* showing the organization of the file. The file was created with `BLOCK-CONTROL-INFO=*NO`.

Example 9: Copying two or more files with a COPY-FILE command

```

/copy-file from=<proc>.<pctest>.*,to=<2>-backup.<1>edure.<3>,
          output=*sysout,dialog-contr=*file-change _____ (1)
% DMS0810 EXECUTE JOB FOR FILE(S) ':20SG:$USER1.PROC.PTEST.EDT' (Y=YES; N=NO; T=
=TERMINATE; ?=HELP/FURTHER OPTIONS)?y
% DMS0816 JOB FOR THE FILE NAME ':20SG:$USER1.PROC.PTEST.EDT' SUCCESSFULLY EXEC
UTED WITH FILE NAME ':20SG:$USER1.PTEST-BACKUP.PROCEDURE.EDT'
% DMS0810 EXECUTE JOB FOR FILE(S) ':20SG:$USER1.PROC.PTEST.KOSTEN' (Y=YES; N=NO
; T=TERMINATE; ?=HELP/FURTHER OPTIONS)?y
% DMS0816 JOB FOR THE FILE NAME ':20SG:$USER1.PROC.PTEST.KOSTEN' SUCCESSFULLY E
XECUTED WITH FILE NAME ':20SG:$USER1.PTEST-BACKUP.PROCEDURE.KOSTEN'
% DMS0810 EXECUTE JOB FOR FILE(S) ':20SG:$USER1.PROC.PTEST.KOSTEN1' (Y=YES; N=N
O; T=TERMINATE; ?=HELP/FURTHER OPTIONS)?y
% DMS0816 JOB FOR THE FILE NAME ':20SG:$USER1.PROC.PTEST.KOSTEN1' SUCCESSFULLY
EXECUTED WITH FILE NAME ':20SG:$USER1.PTEST-BACKUP.PROCEDURE.KOSTEN1'
% DMS0810 EXECUTE JOB FOR FILE(S) ':20SG:$USER1.PROC.PTEST.KOSTEN2' (Y=YES; N=N
O; T=TERMINATE; ?=HELP/FURTHER OPTIONS)?y
% DMS0816 JOB FOR THE FILE NAME ':20SG:$USER1.PROC.PTEST.KOSTEN2' SUCCESSFULLY
EXECUTED WITH FILE NAME ':20SG:$USER1.PTEST-BACKUP.PROCEDURE.KOSTEN2'
% DMS0810 EXECUTE JOB FOR FILE(S) ':20SG:$USER1.PROC.PTEST.REVERSE' (Y=YES; N=N
O; T=TERMINATE; ?=HELP/FURTHER OPTIONS)?y
% DMS0816 JOB FOR THE FILE NAME ':20SG:$USER1.PROC.PTEST.REVERSE' SUCCESSFULLY
EXECUTED WITH FILE NAME ':20SG:$USER1.PTEST-BACKUP.PROCEDURE.REVERSE'
% DMS0810 EXECUTE JOB FOR FILE(S) ':20SG:$USER1.PROC.PTEST.STEUER' (Y=YES; N=NO
; T=TERMINATE; ?=HELP/FURTHER OPTIONS)?y
% DMS0816 JOB FOR THE FILE NAME ':20SG:$USER1.PROC.PTEST.STEUER' SUCCESSFULLY E
XECUTED WITH FILE NAME ':20SG:$USER1.PTEST-BACKUP.PROCEDURE.STEUER'

/copy-file from=prot.*feh1/,to=fehlerprotokoll-/*,output=*sysout _____ (2)
% DMS0810 EXECUTE JOB FOR FILE(S) ':20SG:$USER1.PROT.*FEHL/' (Y=YES; N=NO; T=TE
RMINATE; ?=HELP/FURTHER OPTIONS)?y
% DMS0812 '05A9' DMS ERROR CODE OCCURRED DURING EXECUTION OF THE JOB FOR THE FI
LE NAME ':20SG:$USER1.PROT.SDFFRAME.FEHL1' WITH THE FILE NAME ':20SG:$USER1.FEHL
ERPROTOKOLL-1.SDFFRAME.'. FURTHER INFORMATION: /HELP-MSG DMS05A9
% DMS0813 CONTINUE JOB AFTER ERROR (Y=YES; N=NO; R=RETRY; ?=HELP/FURTHER OPTION
S)?y _____ (3)
% DMS0816 JOB FOR THE FILE NAME ':20SG:$USER1.PROT.VARFEHL1' SUCCESSFULLY EXECU
TED WITH FILE NAME ':20SG:$USER1.FEHLERPROTOKOLL-1.VAR'
% DMS0816 JOB FOR THE FILE NAME ':20SG:$USER1.PROT.VARFEHL2' SUCCESSFULLY EXECU
TED WITH FILE NAME ':20SG:$USER1.FEHLERPROTOKOLL-2.VAR'

```

- (1) All files whose first subname is *PROC* and whose second subname is *PTEST* are to be copied. The newly formed names are to have *PTEST* (second search string in index notation) and *-BACKUP* as the first subname. The second subname is to be made up of *PROC* (first search string) and *EDURE*. The third name part of the search string is to remain unchanged. Copying is to be confirmed for each file (DIALOG-CONTROL) and the new name is to be output to SYSOUT (OUTPUT).
- (2) All files which begin with the subname *PROT*, contain a freely selectable character string and end with the string *FEHL* and another character are to be copied. The new names are to begin with the subname made up of the string *FEHLERPROTOKOLL* (= error log) and the last character of the search string. This is to be followed by the string identified by * in the search string.
- (3) An error occurred when editing the file *PROT.SDFFRAME.FEHL1*. The newly formed name of the target file would end with a period and is therefore not possible. Processing is continued by entering *YES*.

COPY-GS-PARTITION

Copy data of a GS partition

Description status:	GSMAN V19.0A
Functional area:	Global storage administration
Domain:	DEVICE
Privileges:	TSOS

Function

Systems support can use the COPY-GS-PARTITION command to copy data from one partition (source partition) to another, existing partition (target partition) within the global storage (GS) medium.

This function is only available for VIRTGS partitions.

The following combinations of source and target partitions are possible:

Source partition	Target partition	Command processing
Dual partition	Dual partition	A copy operation is performed for each GS unit.
Dual partition	Mono-partition	The data of the GS unit on which the mono-partition is located is copied.
Mono-partition	Dual partition	The data is duplicated in the dual partition.

Note

The command is not accepted if either the source or the target partition is in use. Partition utilization is displayed in the output from the SHOW-GS-STATUS command with ACCESS=NONE.

For more details on administering global storage, see the “Introduction to System Administration” [14].

Format

COPY-GS-PARTITION
FROM-PARTITION-ID = <name 1..8> ,TO-PARTITION-ID = <name 1..8>

Operands**FROM-PARTITION-ID = <name 1..8>**

Name of the GS partition whose data is to be copied (source).

TO-PARTITION-ID = <name 1..8>

Name of the GS partition to which the source partition data is to be copied (target). The target partition must already exist.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
2	0	EGC0230	Target partition may contain inconsistent data
	1	CMD0202	Syntax error
	32	CMD0221	Internal error
	64	CMD0216	Privileges error
	64	EGC0112	No GS available
	64	EGC0220	Partition in use
	64	EGC0221	GS unit not available
	64	EGC0222	Partition does not exist
	64	EGC0223	Error copying partition
	64	EGC0224	Partition is split
	64	EGC0225	Target partition contains invalid data
	64	EGC0226	Not possible to copy from DUAL to MONO partition
	64	EGC0227	Different partition types
	64	EGC0229	Partition is not a VIRTGS partition
	128	EGC0010	GSMAN subsystem is not ready

COPY-JV

Copy job variable

Description status:	JV V15.1A
Functional area:	Job variables
Domain:	JOB-VARIABLES
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SECURITY-ADMINISTRATION SAT-FILE-MANAGEMENT SAT-FILE-EVALUATION

This function is available to the user only if the chargeable software product JV has been loaded as a subsystem.

Function

The COPY-JV command copies the content of a job variable (source JV) to another job variable (target JV). Optionally it is possible to copy not only the content but also the protection attributes of the source JV with the exception of any existing MONJV or CJC protection.

Both permanent and temporary job variables can be copied.

Read access must be permitted for the source JV and write access for the target JV. If a target JV does not exist it is created. When a permanent JV is created the maximum permitted number specified in the user entry may not be exceeded (see also the CREATE-JV command).

Privileged functions

Systems support (TSOS privilege) is by default co-owner of all job variables (and can therefore also create or copy job variables under any user ID). This co-ownership can be restricted for permanent job variables if SECOS is used.

Format

COPY-JV	Alias: CPJV
<pre> FROM-JV = <filename 1..54 without-gen-vers> / *LINK(...) *LINK(...) LINK-NAME = <alphanum-name 1..7> , TO-JV = <filename 1..54 without-gen-vers> / *LINK(...) *LINK(...) LINK-NAME = <alphanum-name 1..7> , PROTECTION = *STD / *SAME , REPLACE-OLD-JV = *YES / *NO </pre>	

Operands

FROM-JV = <filename 1..54 without-gen-vers>

Name of the job variable which is to be copied (source JV).

Write access must be permitted (for a JV under a foreign user ID either USER-ACCESS=ALL-USERS or read permission via BASIC-ACL or GUARDS or co-ownership must exist).

FROM-JV = *LINK(...)

The source JV is named via a link name.

LINK-NAME = <alphanum-name 1..7>

Link name of the JV.

TO-JV = <filename 1..54 without-gen-vers>

Name of the job variable which is to be copied to (target JV).

If the target JV is not yet cataloged, it is created. In this case only the user's own user ID may be specified or a user ID for which the user is co-owner.

If the target JV is cataloged, write access must be permitted (in the case of a JV under a foreign user ID, either the standard access control must exist with USER-ACCESS= *ALL-USERS or write permission via BASIC-ACL or GUARDS or co-ownership). However, the target JV is overwritten only if REPLACE=*YES is specified (default).

TO-JV = *LINK(...)

The target JV is named via a link name.

LINK-NAME = <alphanum-name 1..7>

Link name of the JV.

PROTECTION = *STD / *SAME

Specifies whether the protection attributes of the source JV are also to apply for the target JV.

If MONJV or CJC protection exists for the source JV, this is **not** taken over.

PROTECTION = *STD

The protection attributes are *not* applied to the target JV. If the target JV is to be newly created, it will be assigned the system defaults as protection attributes (see also the defaults of the CREATE-JV command). The existing protection attributes are retained if the target JV already exists.

PROTECTION = *SAME

The target JV is assigned the same protection attributes as the source JV (i.e. for *ACCESS*, *USER-ACCESS*, *OWNER*, *GROUP*, *OTHERS*, *EXPIR-DATE*, *EXPIR-TIME*, *MAN-CLASS*, defined GUARDS, and the same passwords; see also the output fields of the SHOW-JV-ATTRIBUTES command). If MONJV or CJC protection exists it is not taken over. The PROTECTION=*SAME specification is ignored in the following cases (i.e. *STD applies):

- The target JV is a temporary JV.
 - CJC protection exists for the target JV.
 - The target JV resides under a foreign user ID and the caller is not a co-owner.
- As management classes are pubset-specific they can only be taken over if the source and target JVs reside on the same pubset. When copying to another pubset an existing target JV retains its management class; a new target JV is assigned MANAGEMENT-CLASS=*NONE (as with PROTECTION=*STD).

In the case of a source JV which resides under a foreign ID and is protected with BASIC-ACL or GUARDS, the protection attributes USER-ACCESS, BASIC-ACL and GUARDS are set to the system defaults (see the CREATE-JV command).

REPLACE-OLD-JV = *YES / *NO

Specifies whether an existing target JV is to be overwritten.

REPLACE-OLD-JV = *YES

An existing target JV is overwritten without any message being issued.

REPLACE-OLD-JV = *NO

An existing target JV is not overwritten. The command is rejected. In procedures error handling is triggered (spin-off mechanism in non-S procedures or SDF-P error handling in S procedures).

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed
2	0	CMD0001	Command executed with a warning
	1	CMD0202	Syntax error
	32	CMD0221	System error
	64	JVS04E0	Command not executable in the call environment; if possible, remove cause of error (see SYSOUT message JVS04xx)
	130	JVS04E1	Command cannot be executed at this time; for cause see SYSOUT message JVS04xx
	130	CMD2282	Subsystem JV not available for indefinite time

COPY-POSIX-FILE

Copy file from or to POSIX

Description status:	POSIX-BC V10.0A
Functional area:	File processing POSIX administration and application
Domain:	FILE
Privileges:	all privileges

Function

The COPY-POSIX-FILE command offers the functionality of the POSIX command *bs2cp* in BS2000:

- Copying files from the POSIX file system to BS2000 files or elements of a PLAM library
- Copying BS2000 files or elements of a PLAM library to the POSIX file system (library elements are only copied to simple files)

Access to BS2000 files via RFA (Remote File Access) is not supported. For encrypted files, the crypto password must be entered in the crypto password table (see the ADD-CRYPTO-PASSWORD command).

Notes on command execution

The specifications in the COPY-POSIX-FILE command generate a *bs2cp* command with the corresponding parameters (if a list of file names is specified, a corresponding number of *bs2cp* commands) which is called in a POSIX shell (see the START-POSIX-SHELL command). The caller must therefore have a HOME directory in POSIX. The settings from the .profile in the HOME directory (e.g. changing the current directory and values of the environment variables *IO_CONVERSION* and *BS2CPTABS*) affect the copy process.

Before the *bs2cp* command is called, two other POSIX commands may be issued if required:

- The handling of text and binary files when copying (RECORD-CONVERSION operand) is controlled via an *ftyp* command.
- The file attributes of the target files are optionally controlled via a *bs2file* command (FILE-ATTRIBUTES operand).

The POSIX command *bs2cp* is described in the “POSIX Basics for Users and System Administrators” manual [28], the *bs2file* and *ftyp* commands in the “POSIX Commands” manual [29].

The mode of operation of the COPY-POSIX-FILE command is described in detail in the “POSIX Basics for Users and System Administrators” manual [28].

Format

COPY-POSIX-FILE	Alias: CPXF / BS2CP
<p>COPY-DIRECTION = *FROM-POSIX / *TO-POSIX</p> <p>,POSIX-FILE = *BY-SOURCE(...) / list-poss(2000): <posix-pathname 1..1023></p> <p>*BY-SOURCE(...)</p> <ul style="list-style-type: none"> POSIX-DIRECTORY = _ / <posix-pathname 1..1023 without-wild> ,PREFIX = *NONE / <c-string 0..80 with-low> ,SUFFIX = *NONE / <c-string 0..80 with-low> <p>,BS2000-FILE = *BY-SOURCE(...) / *LIBRARY-ELEMENT(...) / list-poss(2000): <filename 1..54 with-wild(80)></p> <p>*BY-SOURCE(...)</p> <ul style="list-style-type: none"> PREFIX = *NONE / <c-string 0..53> ,SUFFIX = *NONE / <c-string 0..40> <p>*LIBRARY-ELEMENT(...)</p> <ul style="list-style-type: none"> LIBRARY = <filename 1..54> ,ELEMENT = *BY-SOURCE(...) / list-poss(2000): <composed-name 1..64 with-under-wild>(…) *BY-SOURCE(...) <ul style="list-style-type: none"> VERSION = *HIGHEST-EXISTING / *UPPER-LIMIT / <composed-name 1..24 with-under> ,PREFIX = *NONE / <c-string 0..63> ,SUFFIX = *NONE / <c-string 0..63> <composed-name 1..64 with-under-wild>(…) VERSION = *HIGHEST-EXISTING / *UPPER-LIMIT / <composed-name 1..24 with-under> ,TYPE = *S / *D / *J / *M / *P / *X / *L <p>,WRITE-MODE = *BY-DIALOG / *REPLACE / *CREATE</p> <p>,CHARACTER-CONVERSION = *NO / *YES(...)</p> <p>*YES(...)</p> <ul style="list-style-type: none"> TABLE = *STD / <posix-pathname 1..1023 without-wild> <p>,OUTPUT = *NONE / *SYSOUT</p> <p>,RECORD-CONVERSION = *TEXT(...) / *BINARY</p> <p>*TEXT(...)</p> <ul style="list-style-type: none"> SUBSTITUTE-TABULATOR = *YES / *NO <p>,FILE-ATTRIBUTES = *STD / *PARAMETER(...)</p> <p>*PARAMETER(...)</p> <ul style="list-style-type: none"> FILE-NAME = *ALL / <filename 1..54> ,ATTRIBUTES = *STD / <c-string 0..1000> 	

Operands

COPY-DIRECTION =

Copy direction.

COPY-DIRECTION = *FROM-POSIX

POSIX files are copied to BS2000.

COPY-DIRECTION = *TO-POSIX

BS2000 files or PLAM elements are copied to POSIX.

POSIX-FILE = *BY-SOURCE / list-poss(2000): <posix-pathname 1..1023>

Specifies the POSIX files to be used during copying.

POSIX-FILE = *BY-SOURCE

*This specification is only possible when copying *TO-POSIX:*

The names of the POSIX files should be derived from the BS2000 names. If multiple BS2000 files are to be copied to POSIX, this operand value must be specified.

POSIX-DIRECTORY =

Specifies the directory to which the BS2000 files or the PLAM elements are to be copied.

POSIX-DIRECTORY = .

The files are copied to the directory currently set.

As default, this is the home directory of the calling BS2000 user. Another current directory can be set by using the change directory command in the *.profile* file (with the POSIX command *cd*).

POSIX-DIRECTORY = <posix-pathname 1..1023 without-wild>

The files are copied to the explicitly specified directory.

PREFIX = *NONE / <c-string 0..80 with-low>

Prefix for the POSIX file name. *NONE is the default, i.e. no prefix is used.

SUFFIX = *NONE / <c-string 0..80 with-low>

Suffix for the POSIX file name. *NONE is the default, i.e. no suffix is use.

POSIX-FILE = list-poss(2000): <posix-pathname 1..1023>

The names of POSIX files are entered explicitly. The following must be taken into account:

- When copying **FROM-POSIX*:
Enter one or more absolute or relative path names of the POSIX files. The POSIX wildcard syntax (shell special characters for file name replacement) is supported.
- When copying **TO-POSIX*:
If only one BS2000 file is copied and the name of the target file is to be explicitly specified, the absolute or relative path name of a POSIX file must be specified. Wildcard syntax is not permitted.
- As default, relative path names refer to the home directory of the calling BS2000 user. Another directory can be set by using the change directory (*cd*) command in the *.profile* file.

BS2000-FILE = *BY-SOURCE(...) / *LIBRARY-ELEMENT(...) /**list-poss(2000): <filename 1..54 with-wild(80)>**

Specifies the BS2000 files or the PLAM elements to be used during copying.

BS2000-FILE = *BY-SOURCE(...)

*This specification is only possible when copying *FROM-POSIX:*

The names of the BS2000 files are derived from the names of the POSIX files. This is a mandatory entry when more than one POSIX file is to be copied to BS2000.

PREFIX = *NONE / <c-string 0..53 with-low>

Prefix for the BS2000 file name. *NONE is the default, i.e. no prefix is used.

SUFFIX = *NONE / <c-string 0..40 with-low>

Suffix for the BS2000 file name. *NONE is the default, i.e. no suffix is use.

BS2000-FILE = *LIBRARY-ELEMENT(...)

PLAM elements are used during copying.

LIBRARY = <filename 1..54>

Name of the PLAM library.

ELEMENT =

Specifies the PLAM elements to be used during copying.

ELEMENT = *BY-SOURCE(...)

*This specification is only possible when copying *FROM-POSIX:*

The element name is to be derived from the POSIX name. This is a mandatory entry if more than one POSIX file is to be copied to a PLAM library in BS2000.

VERSION =

Specifies which version of an element is to be used.

VERSION = *HIGHEST-EXISTING

The target element with the highest version is used. The following must be taken into account:

- If an element does not yet exist, it is assigned version *001*.
- If existing elements are copied, then the element is overwritten with the highest version.

VERSION = *UPPER-LIMIT

The target element should be assigned the highest possible version (X'FF'; this corresponds to the tilde character in the *bs2cp* command).

VERSION = <composed-name 1..24 with-under>

The version is explicitly entered.

PREFIX = *NONE / <c-string 0..63 with-low>

Prefix for the element file name. *NONE is the default, i.e. no prefix is used.

SUFFIX = *NONE / <c-string 0..63 with-low>

Suffix for the element file name. *NONE is the default, i.e. no suffix is used.

ELEMENT = list-poss(2000): <composed-name 1..64 with-under-wild>(…)

The names of the elements are entered explicitly. The following must be taken into account:

- When copying **TO-POSIX*:
Enter one or more element names. The LMS wildcard syntax (“*”, “<”, “:”, “>”) is supported.
In contrast to the POSIX command *bs2cp* a list of element names can also be specified here. In this case the POSIX command *bs2cp* is called for each element name (with/without wildcard). The entries for the other operands then apply for all *bs2cp* calls.
- When copying **FROM-POSIX*:
Explicit specification of an element name (wildcards are not allowed here) if only one POSIX file is copied and the name of the target element is to be defined explicitly.

VERSION =

Specifies the element version used.

VERSION = *HIGHEST-EXISTING

The element with the highest version is used.

VERSION = *UPPER-LIMIT

The copied element should be assigned the highest possible version (X'FF').

VERSION = <composed-name 1..24 with-under>

The version is entered explicitly.

TYPE = *S / *D / *J / *M / *P / *X / *L

Specifies the type of PLAM elements handled. As default, the type S (Source) is used.

BS2000-FILE = list-poss(2000): <filename 1..54 with-wild(80)>

The names of the BS2000 files are entered explicitly. The following must be taken into account:

- When copying **TO-POSIX*:
Enter one or more file names. The BS2000 wildcard syntax is supported with restrictions (only the wildcard “*”).
In contrast to the POSIX command *bs2cp* a list of BS2000 file names can also be specified here. In this case the POSIX command *bs2cp* is called for each file name (with/without wildcard). The entries for the other operands then apply for all *bs2cp* calls.
- When copying **FROM-POSIX*:
Explicit specification of a file name (wildcards are not allowed here) if only one POSIX file is copied and the name of the target file is to be defined.

WRITE-MODE =

*This specification is only relevant when copying *FROM-POSIX.*

Specifies whether target files or elements that already exist in BS2000 are overwritten (this is similar to the *-f* option in the POSIX command *bs2cp*).

WRITE-MODE = *BY-DIALOG

In this case the dialog will display a prompt asking if an existing file should be overwritten.

```
bs2cp: overwrite A ? y (yes), n (no), a (all) or q (quit)
```

WRITE-MODE = *REPLACE

The dialog prompt is suppressed and existing files or elements are always overwritten.

WRITE-MODE = *CREATE

Target files or elements which do not yet exist are created. Existing target files or elements are not overwritten.

CHARACTER-CONVERSION =

Specifies if character conversion can be carried out during the copying process (this is similar to the *-k* and *-t* options in the POSIX *bs2cp* command).

CHARACTER-CONVERSION = *NO

Character conversion is not carried out.

CHARACTER-CONVERSION = *YES(...)

Character conversion is carried out.

TABLE =

Specifies the conversion table.

TABLE = *STD

Internal POSIX default tables are used (this is similar to the *-k* option in the POSIX command *bs2cp*).

TABLE = <posix-pathname 1..1023 without-wild>

The conversion table is entered explicitly (this is similar to the *-t* option in the POSIX command *bs2cp*).

Note

The shell variable *BS2CPTABS* (see *bs2cp*) is not supplied with values via this specification. If required the variable in the *.profile* can be set.

OUTPUT =

Specifies whether extended logging of the *bs2cp* command is to be output (this is similar to the *-l* option in the POSIX command *bs2cp*).

OUTPUT = *NONE

The extended logging is not output.

OUTPUT = *SYSOUT

The extended logging is output to SYSOUT.

RECORD-CONVERSION =

Entry specifying how the contents of BS2000 files are to be handled during copying. This parameter generates the POSIX command *ftyp* with the corresponding parameters. If an operand value is not explicitly specified (corresponds to the preset value RECORD-CONVERSION=*TEXT(SUBSTITUTE-TABULATOR=*YES)), *ftyp text* is set as the default.

RECORD-CONVERSION = *TEXT(...)

SAM files or PLAM library elements are handled as text files. "newline" in the POSIX file becomes a change of record in the BS2000 file (or vice versa, depending on the copying direction).

SUBSTITUTE-TABULATOR =

Specifies how tabulator characters are to be handled.

SUBSTITUTE-TABULATOR = *YES

Tabulator characters are to be filled (*ftyp text*).

SUBSTITUTE-TABULATOR = *NO

Tabulator characters are retained (*ftyp textbin*).

RECORD-CONVERSION = *BINARY

SAM files or PLAM library elements are handled as binary files.

FILE-ATTRIBUTES =

*This specification is only relevant when copying *FROM-POSIX.*

When POSIX files are copied to BS2000 (not for PLAM elements) the file attributes of the target files in BS2000 can be entered in the same way as for the POSIX command *bs2file*. Depending on the parameter entry, a *bs2file* command is issued in the shell before the actual *bs2cp* command.

FILE-ATTRIBUTES = *STD

No *bs2file* command is issued:

Files which do not yet exist are assigned the standard file attributes (SAM access method, variable record format and standard blocking). Existing files retain their attributes. If only the catalog entry exists for a file (no OPEN has taken place), the ISAM access method is used.

FILE-ATTRIBUTES = *PARAMETER(...)

A *bs2file* command is issued when copying. The file attributes are specified in the same way as for the POSIX command *bs2file*.

FILE-NAME =

Specifies the file for which the attributes are to be set.

FILE-NAME = *ALL

The attributes entered are valid for the first file to be copied with any name (corresponds to the "*" entry in the POSIX command *bs2file*).

FILE-NAME = <filename 1..54>

The attributes entered are valid for the first file which is to be copied which has this name.

ATTRIBUTES =

Specifies the file attributes.

ATTRIBUTES = *STD

The standard file attributes are used.

ATTRIBUTES = <c-string 1..1000>

Explicit specification of the file attributes.

The file attributes supported are described for the POSIX command *bs2cp* (see the "POSIX Basics for Users and System Administrators" manual [28]). The entry format for the file attributes (operands and possible values) is described under the FILE macro (see the "DMS Macros" manual [12]).

Example: ATTRIBUTES='FCBTYPE=SAM,RECFORM=F,BLKSIZE=80'

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
	64	POS6010	Invalid combination of the specifications BS2000-FILE=*BY-SOURCE and COPY-DIRECTION=*TO-POSIX
	64	POS6011	Invalid combination of the specifications BS2000-FILE=*LIBRARY-ELEMENT(...,ELEMENT=*BY-SOURCE,...) and COPY-DIRECTION=*TO-POSIX
	64	POS6012	Invalid combination of the specifications POSIX-FILE=*BY-SOURCE and COPY-DIRECTION=*FROM-POSIX
	64	POS6013	More than one BS2000 file specified as target.
	64	POS6014	More than one PLAM library element specified as target.
	64	POS6015	More than one POSIX file specified as target.
	64	POS6016	More than one BS2000 file or PLAM library element specified as source and one POSIX file specified as target; however, this is not a directory.
	64	POS6017	More than one POSIX file specified as source, but BS2000-FILE=*BY-SOURCE not specified.
	64	POS6018	The required version of SDF or SDF-P-BASYS is not installed.
	64	POS6019	Error when starting the POSIX shell.
	64	POS6020	Error when executing the POSIX command bs2cp.
	64	POS6021	Invalid specification of FILE-ATTRIBUTES.
	64	POS6022	Error when executing the POSIX command bs2file.
	64	POS6023	Invalid specification of wildcards.
	64	POS6024	Invalid specification in the ATTRIBUTES operand.

COPY-SYSTEM-FILE

Copy contents of logical system file

Description status:	SYSFILE V19.0A
Functional area:	File processing
Domain:	JOB PROCEDURE PROGRAM
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Function

The COPY-SYSTEM-FILE command copies the contents of the logical system file SYSLST, for which the primary assignment is in force, to a file. The user can thus have the contents stored in a cataloged file without having previously performed a corresponding assignment of SYSLST. The present contents of SYSLST are immediately accessible to the user in the specified file, i.e. there is no need to wait until a listing is available which is only produced at the end of the job or previously by means of PRINT-DOCUMENT. After the copy operation, output to the logical system file is continued.

The part of the logical system file SYSLST that was copied with the COPY-SYSTEM-FILE command **cannot** be deleted. With the DELETE-SYSTEM-FILE command, only the part of the logical system file SYSLST that has been created since the last COPY-SYSTEM-FILE command can be deleted.

Format

COPY-SYSTEM-FILE	Alias: CPSF
FILE-NAME = <u>*SYSLST</u> TO-FILE = <filename 1..54 without-gen-vers>	

Operands**FILE-NAME = *SYSLST**

Name of the logical system file whose contents are to be copied. The default value is the logical system file SYSLST.

TO-FILE = <filename 1..54 without-gen-vers>

Name of the file to which the contents of the system files are to be copied. The specified file is opened with OPEN=OUTPUT, i.e. the contents of any existing file are lost. If insufficient storage space is available for the user ID, the command is rejected.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
	32	SSM3284	DMS error during command execution
	64	SSM3282	System file has no primary assignment
	64	SSM3283	System file is empty

CREATE-ALTERNATE-INDEX

Create secondary index for NK-ISAM file

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processingg
Domain:	FILE
Privileges:	STD-PROCESSING

Function

There is a key field for each record in an NK-ISAM file, which contains the record's primary key. The user can specify the position and length of this primary key in the ADD-FILE-LINK command (KEY-POSITION and KEY-LENGTH operands) when the file is created. These details are copied into the catalog entry (see the command SHOW-FILE-ATTRIBUTES). All primary keys are managed internally by means of a directory, called the primary index. NK-ISAM finds any required record in the file by looking up its primary key in the primary index.

The command CREATE-ALTERNATE-INDEX allows an additional, secondary, index to be set up for the file. It defines an additional key field for the records in the file. For records which already exist (which will be read in sequentially), NK-ISAM creates an internal directory, which is called the secondary index. This is made up of the secondary keys for each record, their corresponding primary keys, and the entries sorted into order of the secondary key values (see the SORT-WORK-FILE operand).

Secondary keys are supported in macro calls for the ISAM access method (see the "DMS Macros" manual [12]). When a particular secondary key value is used for a read or positioning operation within the file, NK-ISAM looks through the entries to find this key. The entry which it finds will contain the corresponding primary key value. Using the primary key determined in this way, the required record is looked up in the primary index.

Any particular secondary key value may be contained in more than one record (depending on the DUPLICATE-KEY operand). When secondary keys are being used, the primary key values must always be unique. The correspondence between the secondary and primary keys then allows the record(s) to be unambiguously identified.

The user may define up to 30 different secondary keys for a file. Each of the secondary keys which is defined, and the associated secondary index which is created for it, has a name which must be unique (see the KEY-NAME operand). Provided that the following requirements are met, the key fields may be in any required position:

- For variable length records, allowance must be made for the 4-byte record length field at the start of the record.
- Any key field which is defined must lie within the record. In the case of variable length records, the length of the shortest records will represent a limit.
- Key fields may not lie within any overflow blocks, which may arise when a K-ISAM file is converted to an NK-ISAM file if the maximum length of record was used.

Secondary keys cannot be managed separately from the primary key. When a file is extended, the secondary indices are automatically extended. For large files it is better (for performance reasons) to set up the required secondary index or indices after the file has been created.

The SHOW-INDEX-ATTRIBUTES command provides information on secondary keys which have been defined, or secondary indices which have been created, for a file. Its execution involves read access to the file (OPEN=INPUT).

The user can delete secondary indices again using the DELETE-ALTERNATE-INDEX command.

Terminating index creation:

- If NK-ISAM detects an error during the creation of an index (e.g. a data record which is too short), any partly created index will be deleted.
- If the index creation is terminated due to a system crash, the file is locked, and can be recreated using the REPAIR-DISK-FILES command. When this is done, the secondary index is deleted.
- An incompletely created index is retained, and is identified as INCOMPLETE if it is output using SHOW-INDEX-ATTRIBUTES. Attempts to open the file will be rejected with DMS0D84. The incomplete secondary index can only be deleted, and then recreated from the beginning.

Format

CREATE-ALTERNATE-INDEX
<pre>FILE-NAME = <filename 1..54> ,ALTERNATE-INDEX = list-poss(30): [*PARAMETERS](...) [*PARAMETERS](...) KEY-NAME = <name 1..8> ,KEY-POSITION = <integer 1..32496> ,KEY-LENGTH = <integer 1..127> ,DUPLICATE-KEY = *YES / *NO ,SORT-WORK-FILE = *STD / *LINK(...) *LINK(...) LINK-NAME = <name 1..8></pre>

Operands

FILE-NAME = <filename 1..54>

Name of the NK-ISAM file.

The file must fulfil the following requirements:

- It must be already cataloged.
- It must be an NK-ISAM file, i.e. the PAM keys are at the beginning of each PAM page (BLK-CONTR=DATA).
- The primary index must already exist, i.e. the file must have already been opened once in OUTPUT mode.
- The file must not contain any primary keys with the same value (duplicate keys).
- The primary keys must not contain any logical or value flags (see the LOGICAL-FLAG-LENGTH and VALUE-FLAG-LENGTH operands in the ADD-FILE-LINK command).

ALTERNATE-INDEX = list-poss(30): *PARAMETERS(...)

For the secondary index which is to be created, this operand specifies the index name, the position and length of the key in the record, and the permissibility of duplicate secondary key values in different records.

Up to 30 secondary indices can be defined in a list.

KEY-NAME = <name 1..8>

Defines the name of the secondary index.

The selected name must be unique. The SHOW-INDEX-ATTRIBUTES command can be used to establish which names have already been used.

KEY-POSITION = <integer 1..32496>

Defines the position of the first byte of the key field within each data record.

Records of *variable* length begin with a 4-byte field, the record length field.

The permitted positions depend on the length of the key field.

KEY-LENGTH = <integer 1..127>

Defines the length of the key field in bytes.

The key field may be up to 127 bytes long. The position of the key field within the record will depend on its position *and* its length. A key field may not lie within an “overflow” block. Overflow blocks are created if a file is converted from K-ISAM format to NK-ISAM format and the maximum record length is used in the K-ISAM format file.

DUPLICATE-KEY = *YES / *NO

Specifies whether the secondary keys in different records may have the same value. The default setting is YES, i.e. duplicate keys are permitted.

In this case, the secondary index may contain several entries for any particular secondary key value. These entries are sorted according to their date of creation (in time stamp order).

SORT-WORK-FILE = *STD / *LINK(...)

Specifies which disk is to be used for the sort run when the secondary index is being created, if main memory provides insufficient space.

When the sort run ends, the assigned file will be *deleted*.

The default setting is *STD, i.e. if it is required, the sort run will use a file named DISWORK.<tsn>.

SORT-WORK-FILE = *LINK(...)

By specifying its file link name, the user can select another file as the work file.

LINK-NAME = <name 1..8>

File link name under which the file to be used is entered in the TFT.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
	32	DMS0A37	Internal system error
	32	DMS0A3A	Inconsistent controll block
	32	DMS0A42	Internal error on opening the file
	32	DMS0A43	Internal error on closing the file
	32	DMS0A44	Error on reading the file
	32	DMS0A45	Error on writing to the file
	32	DMS0A47	DMS error in sort part of NK-ISAM
	32	DMS0A48	Internal error in sort part of NK-ISAM
	32	DMS0A4F	RDTFT error
	64	DMS0A3D	Incomplete secondary key found
	64	DMS0A3F	Secondary index already defined
	64	DMS0A30	Remote BS2000 system has a version < V10.0A
	64	DMS0A31	Specified catalog ID does not exist
	64	DMS0A33	File contains duplicate primary keys
	64	DMS0A34	Secondary index already exists
	64	DMS0A36	File contains ISAM flags
	64	DMS0A39	Invalid KEYPOS parameter
	64	DMS0A40	List includes duplicated names
	64	DMS0A46	Not an NK-ISAM file
	64	DMS0A4D	File contains duplicate secondary key
	64	DMS0A4E	SHARUPD = YES was specified
	128	DMS0A49	Command interrupted
	130	DMS0A32	Specified catalog ID not available
	130	DMS0A38	Insufficient virtual memory
	130	DMS0A3C	Maximum number of secondary keys
	130	DMS0A3E	ISAM pool is overloaded

Examples

Creation of multiple secondary indices

```

/show-file-attr max.file.4,inf=(org=*yes) _____ (1)
%0000000126 :20SG:$USER1.MAX.FILE.4
%----- ORGANIZATION -----
% FILE-STRUC = ISAM          BUF-LEN = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE) = READ-WRITE    IO(PERF) = STD            DISK-WRITE = IMMEDIATE
% REC-FORM = (V,N)          REC-SIZE = 0
% KEY-LEN = 8               KEY-POS = 5
% AVAIL = *STD
% WORK-FILE = *NO          F-PREFORM = *K          SO-MIGR = *ALLOWED
%:20SG: PUBLIC:           1 FILE RES=          126 FRE=          5 REL=          3 PAGES

/cre-alter-index file-name=max.file.4,
                  alter-index=( (key-name=strasse,key-pos=30,key-len=20), _____ (2)
                                (key-name=ort,key-pos=50,key-len=25) ) _____ (3)

/show-index-at max.file.4 _____ (4)
%FILE-NAME: :20SG:$USER1.MAX.FILE.4
% KEY-NAME KEY-POSITION KEY-LENGTH DUPKEY
%-----
% STRASSE 30 20 YES
% ORT 50 25 YES

```

- (1) Returns information on the organization of the NK-ISAM file *MAX.FILE.4*. The primary key begins at the 5th byte (after a record length field of 4 bytes) and is 8 bytes long. The file contains a customer list, where the 8-digit customer number serves as the primary key.

Two secondary indices are defined in a list in the CREATE-ALTERNATE-INDEX command (see points 2 and 3):

- (2) In order to enable access in a program via the street name which is contained in each record, a secondary index with the name *STREET* is to be created for this data field. The data field begins at the 30th byte and is 20 bytes long. Since a street name may be contained in several records, DUPLICATE-KEY=*YES is permitted (default value).
- (3) In order to enable access in a program via the city name which is contained in each record, a secondary index with the name *CITY* is to be created for this data field. The data field begins at the 50th byte and is 25 bytes long. Since a city name may be included in several records, DUPLICATE-KEY=*YES is permitted (default value).
- (4) The SHOW-INDEX-ATTRIBUTES command returns information on all secondary indices that were created for the file *MAX.FILE.4*.

CREATE-DUMP

Generate user dump

Description status: BS2000 OSD/BC V10.0A
Functional area: Program control
Domain: PROGRAM
Privileges: all privileges bar
HARDWARE-MAINTENANCE
OPERATING

Function

The CREATE-DUMP command allows users to initiate creation of a user dump. A dump is only generated if Y is entered in response to the following query:

```
IDA0N45 DUMP DESIRED? REPLY (Y=USER-/AREADUMP TO DISK;  
Y,<VOLUMETYPE>=USER-/AREADUMP TO TAPE; Y,SYSTEM=SYSTEMDUMP; N=NO)
```

Users can convert the user dump to a system dump by setting their read authorization so that the value m is greater than or equal to 3 in the command MODIFY-TEST-OPTIONS PRIVILEG=PARAMETERS(READ=m). This option is only open to users who have the appropriate authorization in the user entry. The SHOW-USER-ATTRIBUTES command displays information about the user entry.

Whether or not the CREATE-DUMP command results in creation of a dump depends on the setting of the test option DUMP (see the MODIFY-TEST-OPTIONS command):

DUMP=*NO: Dump creation is suppressed
DUMP=*STD: Dump creation is suppressed in batch and procedure mode. In dialog mode, a dump is only created if you respond with "Y" to message IDA0N45.
DUMP=*YES: Dump creation is initiated without prompting.
DUMP=*SYSTEM: The user dump is converted into a system dump without prompting.

Format

CREATE-DUMP

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
	32	CMD0221	System error

CREATE-FILE

Define name and attributes for new file

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING TSOS SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION
Routing code:	\$ (bei NBCONOPI=N) bzw. E (bei NBCONOPI=Y)

Function

The CREATE-FILE command creates a catalog entry for a file. The following attributes are defined for the file and stored in the catalog under the specified file name:

- name (FILE-NAME operand)
- type of volume and performance attributes (SUPPORT operand)
- protection attributes (PROTECTION operand)
- method and frequency with which the file is automatically backed up by ARCHIVE or HSMS (SAVE operand)
- user information (USER-INFORMATION and ADM-INFORMATION operands)
- lock against migration when public storage space is automatically managed by the software product HSMS (MIGRATE operand)
- character set (CODED-CHARACTER-SET operand)

Further attributes, see [“Overview of functions” on page 2-384](#).

Privileged functions

The following functions are available only if physical allocation of public space is allowed:

- explicit specification (values other than the default, *STD) of a volume or volume set in the VOLUME or VOLUME-SET operand in the structure SUPPORT=*PUBLIC-DISK (STORAGE-CLASS=*NONE(...)) when creating files on the pubset's disks
- absolute space reservation using SPACE=*ABSOLUTE(...) in the structure SUPPORT=*PUBLIC-DISK(...) (note that no permission is required when SUPPORT=*PRIVATE-DISK)
- intensified migration locking using MIGRATE=*FORBIDDEN

Exception: These functions are also available for work files when no authorization for physical allocation exists.

Physical allocation is allowed in the following cases:

- for tasks with TSOS privilege
- for files of a user ID which has PHYSICAL-ALLOCATION=*ALLOWED in its pubset-specific user entry (see the SHOW-USER-ATTRIBUTES command)
- for files in SF pubsets which have an MRSCAT entry that allows physical allocation for all users (see the SHOW-MASTER-CATALOG-ENTRY command, INFORMATION=*USER) operand.

Systems support (TSOS privilege) personnel can supplement the user information in the file catalog with one to eight bytes of information about the file (ADM-INFORMATION operand).

Systems support personnel (TSOS privilege) can have accesses to the file monitored (AUDIT operand). To configure the AUDIT function, nonprivileged users need appropriate authorization in the user entry of the pubset in which the file is to be created (see the *AUDIT* output field of the SHOW-USER-ATTRIBUTES command).

By default, systems support (TSOS privilege) is a co-owner of all the files (and can therefore create files under all user logons). When SECOS is used, this co-ownership can be restricted.

Systems support can generate files under any user ID (TSOS privilege). In conjunction with the SECOS software product a user can allow other user IDs to act as co-owners. Co-owners are thus allowed to create permanent files under that ID.

Temporary files

Since temporary files are job-related, it is not possible to define file protection for them, i.e. the specified protection attributes (PROTECTION operand) cannot differ from the defaults (except for DESTROY-BY-DELETE, AUDIT, and SPACE-RELEASE-LOCK).

A nonprivileged user can only generate temporary files on the default pubset of his user ID.

Files in SM pubsets

The storage location for a file on an SM pubset can be selected on the basis of a storage class. This simplifies the automatic management of storage space (for details see the “Introduction to System Administration” [14] or the “System-Managed Storage” manual [45]). Systems support personnel allocates to a storage class certain file attributes which are implicitly set for all files assigned to that storage class. If a storage class is specified, there is no need to explicitly specify values for the WORK-FILE, VOLUME-SET, VOLUME, DEVICE-TYPE, IO-ATTRIBUTES, DISK-WRITE, FILE-PREFORMAT and AVAILABILITY operands. The SHOW-STORAGE-CLASS command allows users to find out which storage classes of the SM pubset are available to them and which file attributes are set.

Physical allocation in SM pubsets

A user who is allowed to perform physical allocation of public storage space (see “Privileged functions”) can directly specify a volume or volume set (VOLUME or VOLUME-SET operand) in the STORAGE-CLASS=*NONE(...) structure in order to select a specific volume set of an SM subset as the storage location for a file.

The WORK-FILE attribute is specified implicitly. In addition, a lock on migration to another volume set of the SM subset is placed on the file (S0-MIGRATION=*FORBIDDEN; see also the identically named operand in the this command description).

When S0-MIGRATION=*FORBIDDEN is set (in this case implicitly), the file’s PERFORMANCE and AVAILABILITY attributes are matched to the identically named attributes of the volume set.

These attributes correspond to the levels of PERFORMANCE (STD, HIGH, VERY-HIGH) and AVAILABILITY (STD, HIGH) which are hierarchically arranged in the user catalog. To ensure that there are adequate levels of performance and availability, files with the S0-MIGRATION=*FORBIDDEN attribute are always assigned at least the PERFORMANCE and AVAILABILITY levels defined for the volume set (you can list the attributes of the volume set with the SHOW-PUBSET-DEFINITION-FILE command (and also with SHOW-PUBSET-CONFIGURATION)).

Any explicit specification of the PERFORMANCE and AVAILABILITY operands which selects a lower level in the hierarchy is ignored. The minimum value is recorded in the file’s catalog entry, and the level in the user catalog is set accordingly. Note that a file’s PERFORMANCE attribute may be higher than the maximum permissible value as set in the user entry (*DMS-TUNING-RESOURCES* output field). Then, regardless of the level that has been set, the file will at best be processed using the value which is in the user entry at the time of OPEN processing. The level settings and the value of the DMS-TUNING-RESOURCES attribute can be viewed in the output of the SHOW-USER-ATTRIBUTES command, INFORMATION=*PUBSET-ATTRIBUTES operand.

Example

In conjunction with physical allocation, the user specifies PERFORMANCE=*HIGH or *STD (or this applies implicitly, since it is the default), but the attribute of the chosen volume set is PERFORMANCE=*VERY-HIGH. The file will then automatically be assigned the attribute PERFORMANCE=*VERY-HIGH. If the user entry includes DMS-TUNING-RESOURCES=*CONCURRENT-USE, the file will actually only be processed with a setting of PERFORMANCE=*HIGH.

Work files

The user can create a file as a work file in an SM subset (operand setting WORK-FILE=*YES). The file is created as a permanent file in a volume set defined for that purpose. Systems support personnel can delete work files again at times which it defines.

Files on tapes and tape cartridges

When the catalog entries for tape files are being accessed or created, the special features associated with this storage medium must be taken into account.

For files with standard labels, details of the shareability (USER-ACCESS), access type (ACCESS) and passwords are transferred from the catalog entry to the file labels when the file is created (OPEN).

When a file is created, its file attributes are transferred to the file labels without being checked. Write protection (ACCESS=*READ) can thus be defined for a file which has still to be created. The file can then be opened as an output file, and created; the write protection will take effect after this.

If a file was cataloged using CREATE-FILE, by default it will be shareable unless USER-ACCESS=*OWNER-ONLY is specified in a CREATE-FILE command (or later by a MODIFY-FILE-ATTRIBUTES command before the file is opened for the first time).

Files on remote systems

If a file is created via an RFA connection on a remote system which has its FARMMSAV system parameter set to 1, the catalog entry is marked as modified (its internal version number is given a value of 1) and will if necessary be backed up in the next incremental backup.

Overview of functions

	Function / Meaning	Level 1 operands	Level 2/3 operands
2-389	Specify name of catalog entry	FILE-NAME	
2-390	Specify the volume	SUPPORT	
	Volume: pubset and Net-Storage	= *PUBLIC-DISK	
2-390	– Specify storage type		STORAGE-TYPE
2-390	– File type (on Net-Storage)		FILE-TYPE
2-391	– Define storage class		STORAGE-CLASS
2-391	– Define work file attribute		WORK-FILE
2-392	– Performance requirements		IO-ATTRIBUTES
	– Performance attribute		PERFORMANCE
	– type of I/O operations		USAGE
2-393	– Define time when data consistency is required after write operations		DISK-WRITE
2-395	– preferred file format		FILE-PREFORMAT
2-394	– Define availability requirements		AVAILABILITY
2-395	– Specify volume set		VOLUME-SET
2-396	– Specify the volume		VOLUME
2-396	– Specify device type		DEVICE-TYPE
2-397	– Storage space allocation		SPACE
2-399	– HSMS storage management class		MANAGEMENT-CLASS
2-400	– user information		USER-INFORMATION
2-400	– systems support information		ADM-INFORMATION
2-400	Volume: private disk	= *PRIVATE-DISK	
2-400	– Request volume(s)		VOLUME
2-400	– Specify device type		DEVICE-TYPE
2-401	– Storage space allocation		SPACE
2-403	– Separate storage of data/index for ISAM files		DATA-SUPPORT
2-405	Volume: tape	= *TAPE	
2-405	– Request volume(s)		VOLUME
2-405	– Specify device type		DEVICE-TYPE
2-406	– Request tape device		PREMOUNT-LIST

Table 32: Overview of CREATE-FILE command functions (Part 1 of 2)

	Function / Meaning	Level 1 operands	Level 2/3 operands
2-407	Define protection attributes	PROTECTION = *PARAMETERS	
2-408	– Import protection attributes from file		PROTECTION-ATTR
2-409	– Access rights		ACCESS
2-409	– Shareability		USER-ACCESS
2-410	– Basic access control list		BASIC-ACL
2-412	– Password protection		PASSWORD
2-413	– Define passwords		WRITE-PASSWORD/ READ-PASSWORD / EXEC-PASSWORD
2-415	– Physical deletion (overwrite with binary zeros)		DESTROY-BY-DELETE
2-416	– Monitoring of DMS accesses via SAT/System Exit Routines		AUDIT
2-417	– Lock to prevent release of reserved storage space (disk files only)		SPACE-RELEASE-LOCK
2-417	– Date when file is freed for deletion		FREE-FOR-DELETION
2-417	<i>Only for files on disk and Net-Storage:</i> Type and frequency of automatic data backup by ARCHIVE or HSMS	SAVE	
2-418	– Backup frequency		BACKUP-CLASS
2-418	– Scope of backup		SAVED-PAGES
2-419	<i>Only for files on disk and Net-Storage:</i> Whether the file can be migrated if the software product HSMS is used (automatic management of public storage space) Migration allowed Migration lock Intensified migration lock	MIGRATE = *ALLOWED = *INHIBITED = *FORBIDDEN	
2-419	Code table (XHCS)	CODED-CHARACTER-SET	
2-420	Error handling if the file already exists Normal error handling Suppress error handling	SUPPRESS-ERRORS = *NONE = *FILE-EXISTING	

Table 32: Overview of CREATE-FILE command functions (Part 2 of 2)

Format

CREATE-FILE	Alias: CRF
<p>FILE-NAME = <filename 1..54 without-gen></p> <p>SUPPORT = *PUBLIC-DISK (...) / *PRIVATE-DISK(...) / *TAPE(...) / *NONE</p> <p>*PUBLIC-DISK(...)</p> <p> STORAGE-TYPE = *STD / *PUBLIC-SPACE / *NET-STORAGE(...)</p> <p> *NET-STORAGE(...)</p> <p> FILE-TYPE = *STD / *BS2000 / *NODE-FILE</p> <p> STORAGE-CLASS = *STD / <composed-name 1..8> / *NONE(...)</p> <p> *NONE(...)</p> <p> WORK-FILE = *STD / *NO / *YES</p> <p> IO-ATTRIBUTES = *STD / [*PARAMETERS](...)</p> <p> [*PARAMETERS](...)</p> <p> PERFORMANCE = *STD / *HIGH / *VERY-HIGH / *USER-MAXIMUM</p> <p> USAGE = *READ-WRITE / *WRITE / *READ</p> <p> DISK-WRITE = *STD / *IMMEDIATE / *BY-CLOSE</p> <p> AVAILABILITY = *STD / *HIGH</p> <p> FILE-PREFORMAT = *BY-PUBSET-DEFAULT / *K / *NK2 / *NK4</p> <p> VOLUME-SET = *STD / *CONTROL-VOLUME-SET / <cat-id 1..4></p> <p> VOLUME = *STD / list-poss(255): <vsn 1..6></p> <p> DEVICE-TYPE = *BY-VOLUME / <device></p> <p> SPACE = *STD / *RELATIVE(...) / *ABSOLUTE(...)</p> <p> *RELATIVE(...)</p> <p> PRIMARY-ALLOCATION = <integer 1..2147483647></p> <p> SECONDARY-ALLOCATION = *STD / <integer 0..32767></p> <p> *ABSOLUTE(...)</p> <p> FIRST-PAGE = <integer 1..2147483647></p> <p> SIZE = <integer 1..2147483647></p> <p> MANAGEMENT-CLASS = *NONE / <composed-name 1..8></p> <p> USER-INFORMATION = *NONE / <c-string 1..8 with-low></p> <p> ADM-INFORMATION = *NONE / <c-string 1..8 with-low></p>	

(Part 1 of 4)

***PRIVATE-DISK(...)**

VOLUME = [***ANY**] (...) / list-poss(255): <alphanum-name 1..6>

[***ANY**](...)

| **NUMBER-OF-DEVICES** = 1 / <integer 1..9>

,**DEVICE-TYPE** = ***BY-VOLUME-CATALOG** / <device>

,**SPACE** = ***STD** / ***RELATIVE(...)** / ***ABSOLUTE(...)**

***RELATIVE(...)**

| **PRIMARY-ALLOCATION** = <integer 1..2147483647>

| ,**SECONDARY-ALLOCATION** = ***STD** / <integer 0..32767>

***ABSOLUTE(...)**

| **FIRST-PAGE** = <integer 1..2147483647>

| ,**SIZE** = <integer 1..2147483647>

,**DATA-SUPPORT** = ***SAME** / [***PARAMETERS**](...)

[***PARAMETERS**](...)

| **DATA-DEVICE-TYPE** = <device>

| ,**DATA-VOLUME** = list-poss(255): <alphanum-name 1..6>

| ,**DATA-SPACE** = ***RELATIVE (...)** / ***ABSOLUTE(...)**

| ***RELATIVE(...)**

| | **PRIMARY-ALLOCATION** = <integer 1..2147483647>

| | ,**SECONDARY-ALLOCATION** = ***STD** / <integer 0..32767>

| ***ABSOLUTE(...)**

| | **FIRST-PAGE** = <integer 1..2147483647>

| | ,**SIZE** = <integer 1..2147483647>

***TAPE(...)**

VOLUME = ***NO** / [***ANY**](...) / list-poss(255): <alphanum-name 1..6>

[***ANY**](...)

| **NUMBER-OF-DEVICES** = 1 / <integer 1..9>

,**DEVICE-TYPE** = ***BY-VOLUME-CATALOG** / <device>

,**PREMOUNT-LIST** = ***NONE** / list-poss(255): <integer 0..255>

(Part 2 of 4)

```

,PROTECTION = *STD / [*PARAMETERS](...)
  [*PARAMETERS](...)
    PROTECTION-ATTR = *BY-DEF-PROT-OR-STD / *STD / *FROM-FILE(...)
      *FROM-FILE(...)
        | FILE-NAME = <filename 1..54 without-gen>
,ACCESS = *BY-PROTECTION-ATTR / *WRITE / *READ
,USER-ACCESS = *BY-PROTECTION-ATTR / *OWNER-ONLY / *ALL-USERS / *SPECIAL
,BASIC-ACL = *BY-PROTECTION-ATTR / *NONE / *STD / [*PARAMETERS](...) / *GROUP-X /
  *GROUP-RX / *GROUP-WRX / *ALL-X / *ALL-RX / *ALL-WRX
  [*PARAMETERS](...)
    OWNER = *NO-ACCESS / [*PARAMETERS](...)
      [*PARAMETERS](...)
        | READ = *NO / *YES
        | ,WRITE = *NO / *YES
        | ,EXEC = *NO / *YES
      ,GROUP = *NO-ACCESS / [*PARAMETERS](...)
        [*PARAMETERS](...)
          | READ = *NO / *YES
          | ,WRITE = *NO / *YES
          | ,EXEC = *NO / *YES
        ,OTHERS = *NO-ACCESS / [*PARAMETERS](...)
          [*PARAMETERS](...)
            | READ = *NO / *YES
            | ,WRITE = *NO / *YES
            | ,EXEC = *NO / *YES
      ,GUARDS = *BY-PROTECTION-ATTR / *NONE / [*PARAMETERS](...)
        [*PARAMETERS](...)
          | READ = *NONE / <filename 1..18 without-cat-gen-vers>
          | ,WRITE = *NONE / <filename 1..18 without-cat-gen-vers>
          | ,EXEC = *NONE / <filename 1..18 without-cat-gen-vers>

```

(Part 3 of 4)

```

,WRITE-PASSWORD = *BY-PROT-ATTR-OR-NONE / *NONE / *SECRET / <c-string 1..4> /
                <x-string 1..8> / <integer -2147483648..2147483647>

,READ-PASSWORD = *BY-PROT-ATTR-OR-NONE / *NONE / *SECRET / <c-string 1..4> /
                <x-string 1..8> / <integer -2147483648..2147483647>

,EXEC-PASSWORD = *BY-PROT-ATTR-OR-NONE / *NONE / *SECRET / <c-string 1..4> /
                <x-string 1..8> / <integer -2147483648..2147483647>

,DESTROY-BY-DELETE = *BY-PROTECTION-ATTR / *NO / *YES

,AUDIT = *STD / *NONE / *SUCCESS / *FAILURE / *ALL

,SPACE-RELEASE-LOCK = *BY-PROTECTION-ATTR / *NO / *YES

,FREE-FOR-DELETION = *NONE / <date> / <integer 0..99999>

,SAVE = *STD / *NO / [*PARAMETERS](...)
      [*PARAMETERS](...)
      |
      | BACKUP-CLASS = *STD / *A / *B / *C / *D / *E
      |
      | SAVED-PAGES = *COMPLETE-FILE / *MODIFIED-PAGES

,MIGRATE = *STD / *ALLOWED / *INHIBITED / *FORBIDDEN

,CODED-CHARACTER-SET = *USER-DEFAULT / *NONE / <name 1..8>

,SUPPRESS-ERRORS = *NONE / *FILE-EXISTING

```

(Part 4 of 4)

Operands

FILE-NAME = <filename 1..54 without-gen>

The name to be given to the new file.

You may specify only your own user logon or a user logon to which co-ownership has been assigned.

The maximum number of files specified for the user in the user catalog must not be exceeded. The permitted number of permanent and temporary files may also be limited. The applicable values can be determined from the output of the SHOW-USER-ATTRIBUTES command. A nonprivileged user can only generate temporary files on the default subset of his user ID.

SUPPORT = *PUBLIC-DISK(...)

Type of volume.

SUPPORT = *PUBLIC-DISK(...)

The file is to be created on public disk or Net-Storage. The pubset on which the file is to be created is uniquely identified by the (explicitly specified or default) catalog ID in the file name.

STORAGE-TYPE = *STD / *PUBLIC-SPACE / *NET-STORAGE(...)

Determines the storage type for the file's storage location.

STORAGE-TYPE = *STD

The file is created on the default storage type for storing files on this pubset.

STORAGE-TYPE = *PUBLIC-SPACE

The file is created on the disks of the pubset.

In addition to the entry in the file catalog, the file is also created physically with the smallest possible amount of storage space (see the SPACE operand on [page 2-397](#)).

STORAGE-TYPE = *NET-STORAGE(...)

The file is created on a Net-Storage volume.

The authorization NET-STORAGE-USAGE=*ALLOWED (default) must be entered in the user entry. Catalog entries are generated both on the local pubset and in the relevant file system of the Net-Server. However, the file is not created physically. If no volume is specified, the file is created on the default Net-Storage volume. If no default Net-Storage volume is assigned to the pubset but only one or more explicitly specified Net-Storage volumes, the system determines the Net-Storage volume on which the file is created.



Files with a PAM key, file generation groups, work files and temporary files cannot be created on Net-Storage.

FILE-TYPE = *STD / *BS2000 / *NODE-FILE

Determines the file type of the file to be created.

FILE-TYPE = *STD

The file is created as a BS2000 file on a Net-Storage volume. When the volume of a private disk is specified under STORAGE-CLASS=*NONE(...), the file is created on the private disk.

FILE-TYPE = *BS2000

The file is created as a BS2000 file on a Net-Storage volume. When under STORAGE-CLASS=*NONE(...) specifications regarding the volume or device type are provided which do not permit this, the command is aborted with an error.

FILE-TYPE = *NODE-FILE

The file is created as a node file on a Net-Storage volume. When under STORAGE-CLASS=*NONE(...) specifications regarding the volume or device type are provided which do not permit this, the command is aborted with an error.

STORAGE-CLASS = *STD / <composed-name 1..8> / *NONE(...)

Governs the properties of the file's storage location.

Systems support staff can define storage classes to facilitate automatic management of storage space in SM pubsets. A storage class is allocated certain file attributes which are implicitly set for all files assigned to that storage class. If a storage class is specified, there is no need to explicitly specify values for the WORK-FILE, VOLUME-SET, VOLUME, DEVICE-TYPE, IO-ATTRIBUTES, DISK-WRITE, FILE-PREFORMAT and AVAILABILITY operands.

The SHOW-STORAGE-CLASS command allows users to find out which storage classes of an SM pubset are available to them and which file attributes are set.

Assigning a storage class simplifies the automatic management of storage space in an SM pubset (for details see the "Introduction to System Administration" [14] or the "System-Managed Storage" manual [45]).

STORAGE-CLASS = *STD

A file in an SM pubset is assigned the default storage class from the user entry for that pubset. If the file is to be created in an SF pubset or if there is no default storage class defined, the file is given the same attributes as with STORAGE-CLASS=*NONE with default values.

STORAGE-CLASS = <composed-name 1..8>

A file in an SM pubset is assigned the specified storage class. The storage class must exist and be accessible to the user at the time when the assignment is made. This specification is ignored for a file in an SF pubset; the file is then given the same attributes as with STORAGE-CLASS=*NONE with default values.

STORAGE-CLASS = *NONE(...)

This value has the same effect as *STD if all the following conditions are met:

- The file is created on a volume set intended for permanent storage.
- A default storage class is assigned to the user ID at the SM pubset in question.
- Physical allocation is not permitted.

Only in this case are the operands WORK-FILE, PERFORMANCE, USAGE, DISK-WRITE and AVAILABILITY ignored.

In all other cases, the properties of the file's storage location are determined on the basis of the following operands and no storage class is assigned to the file.

WORK-FILE = *STD / *NO / *YES

The operand is evaluated only for files in SM pubsets. Specifies whether the file is a work file which systems support is allowed to delete at a time which it defines.

WORK-FILE = *STD

The file is to be a work file only if it is stored in a volume set for work files as a result of the value of the VOLUME or VOLUME-SET operand.

WORK-FILE = *NO

The file is not to be a work file.

WORK-FILE = *YES

The file is to be a work file.

IO-ATTRIBUTES = *STD / *PARAMETERS(...)

Specifies the performance requirements demanded by the user for input/output operations. Whether such requirements are satisfied and the extent to which they are fulfilled depends on the cache medium that is defined for the associated pubset (see the output of the SHOW-MASTER-CATALOG-ENTRY command). If the file is created in an SM pubset, the performance requirements are taken into account when the volume set is selected.

Read and write access is performed through a fast cache. The number of disk accesses and the access times are reduced.

Performance attributes for file processing can also be defined in the TFT (see also the ADD-FILE-LINK command).

IO-ATTRIBUTES = *STD

The file has no special performance attributes and is therefore not processed via a cache.

IO-ATTRIBUTES = *PARAMETERS(...)

Performance attributes are determined by the values of the PERFORMANCE and USAGE operands.

PERFORMANCE = *STD / *HIGH / *VERY-HIGH / *USER-MAXIMUM

Specifies the file's performance attribute. It indicates the priority required for the I/O operations selected in the USAGE operand. The highest performance attribute allowed is defined in the user entry (see also the *DMS-TUNING-RESOURCES* output field in the output of the SHOW-USER-ATTRIBUTES command).

PERFORMANCE = *STD

The file has no special performance attribute and is therefore not processed via a cache.

The USAGE operand does not affect the processing of the file in this case.

PERFORMANCE = *HIGH

The file is to be processed through a cache (high performance priority). This specification is only possible for users who have the DMS tuning privilege CONCURRENT-USE or EXCLUSIVE-USE for the pubset (see the output field *DMS-TUNING-RESOURCES* in the output of the SHOW-USER-ATTRIBUTES command).

PERFORMANCE = *VERY-HIGH

The file is to be processed through a cache. The referenced data in the file is to be kept permanently in the cache (highest performance priority; only available in GS). The cache data is only evicted from the cache when the file is closed. Only users who have the DMS tuning privilege EXCLUSIVE-USE for the pubset can specify (see the output for the SHOW-USER-ATTRIBUTES command, output field *DMS-TUNING-RESOURCES*).

Note All cache segments used for this file are locked until the file is closed. If there are many files open with this attribute at the same time, the cache memory available for normally cached files may as a result be so greatly restricted that data accesses to such files can no longer be implemented with an adequate level of performance.

PERFORMANCE = *USER-MAXIMUM

The file is assigned the highest performance attribute contained in the user catalog for the user.

USAGE = *READ-WRITE / *WRITE / *READ

Specifies the I/O operations for which enhanced performance (caching) is required. The default setting is *READ-WRITE, i.e. the requirements apply to read and write operations.

If the file does not have a particular performance attribute (PERFORMANCE=*STD) and if the cache area for the pubset is not defined in such a way that all existing files are served (see the operand CACHED-FILES=*ALL for the MODIFY-PUBSET-CACHING-ATTRIBUTES command), then the USAGE operand has no effect on processing.

USAGE = *READ-WRITE

The performance requirements apply both to read and to write operations.

USAGE = *WRITE

The performance requirements apply to write operations only.

USAGE = *READ

The performance requirements apply to read operations only.

Note With USAGE=*READ-WRITE or *WRITE, write caching is performed only if the conditions for the DISK-WRITE file attributes are met (see CREATE-FILE or MODIFY-FILE-ATTRIBUTES). For write caching with DISK-WRITE= *IMMEDIATE, the cache medium being used must be fail-safe (see the *CACHE-MEDIUM* output field of the SHOW-MASTER-CATALOG-ENTRY command with INFORMATION=*USER).

DISK-WRITE = *STD / *IMMEDIATE / *BY-CLOSE

Specifies the time at which the data must be on a nonvolatile medium (disk or fail-safe cache medium) after a write operation. If a volatile cache medium is used as the write cache, data in the file will not be in a consistent state until CLOSE

processing has been completed because the write data is written out to disk during CLOSE processing. System errors during the processing phase may lead to inconsistencies. Write caching for files containing data which cannot easily be restored should be performed using only fail-safe cache media, i.e. data consistency after each write operation should be requested with DISK-WRITE=*IMMEDIATE.

DISK-WRITE = *STD

The default setting STD is equivalent to the value *IMMEDIATE for permanent files and the value *BY-CLOSE for temporary files.

DISK-WRITE = *IMMEDIATE

Data contained in the file must be on a nonvolatile medium immediately after a write operation.

Write caching is performed for files marked in this way only if a nonvolatile cache area is active for the pubset (see the *CACHE-MEDIUM=NONVOLATILE* output field of the SHOW-MASTER-CATALOG-ENTRY command with INFORMATION= *USER). If a volatile cache area is active for the pubset (*CACHE-MEDIUM=VOLATILE* output field), only read accesses are buffered for the file; write accesses are performed directly on the disks.

DISK-WRITE = *BY-CLOSE

Data contained in the file needs to be on a nonvolatile medium only after CLOSE processing.

Write caching for files marked in this way is performed regardless of whether the cache medium is fail-safe, i.e. even on volatile cache media.

Unsaved cached data is written out during CLOSE processing. The data in the file will not be on a non-volatile medium (disk storage) until that point.

This attribute is suitable only for files containing data that can easily be restored (list files generated during compilation, for example), as a system error may cause inconsistencies in the files.

AVAILABILITY = *STD / *HIGH

Specifies availability requirements for the file. Files for which high availability is required must always be created by the system on suitable volumes (such as a disk with DRV mirroring). If no appropriate volume is available, the command is rejected.

AVAILABILITY = *STD

The file is not subject to special availability requirements.

AVAILABILITY = *HIGH

The file is to have high availability. If the volume is not capable of meeting the requirement, or if no suitable volume set is available within the SM pubset, the command is rejected.

The command is also rejected if applied to a work file or a temporary file.

FILE-PREFORMAT = *BY-PUBSET-DEFAULT / *K / *NK2 / *NK4

The operand is evaluated only for files in SM pubsets. Governs the preferred file format, which is taken into account in the choice of volume set. As long as the file has not yet been opened (no CREATION-DATE in the catalog entry), the chosen storage location is provisional and may still change in the course of OPEN processing.

FILE-PREFORMAT = *BY-PUBSET-DEFAULT

The preferred file format which applies is the format that systems support has defined as the pubset-specific default for the SM pubset using the MODIFY-PUBSET-SPACE-DEFAULTS command. The value set can be determined with the SHOW-PUBSET-SPACE-DEFAULTS command (and also with SHOW-PUBSET-CONFIGURATION).

FILE-PREFORMAT = *K

The file is to be created as a K file (BLOCK-CONTROL-INFO=*PAMKEY).

FILE-PREFORMAT = *NK2

The file is to be created as an NK file in 2K format (BLOCK-CONTROL-INFO=*WITHIN-DATA-2K-BLOCK or *WITHIN-DATA-BLOCK for a SAM file).

FILE-PREFORMAT = *NK4

The file is to be created as an NK file in 4K format (BLOCK-CONTROL-INFO=*WITHIN-DATA-4K-BLOCK or *WITHIN-DATA-BLOCK for a SAM file).

VOLUME-SET = *STD / *CONTROL-VOLUME-SET / <cat-id 1..4>

The specification is relevant only for a file on an SM pubset.

Determines the volume set on which the file is to be created.

Systems support can specify a volume set explicitly using *CONTROL-VOLUME-SET or <cat-id 1..4>.

Nonprivileged users cannot specify a volume set explicitly unless they have authorization to perform physical allocation of public storage space (see [“Privileged functions” on page 2-380](#)).

VOLUME-SET = *STD

The file is stored in a suitable volume set of the SM pubset.

VOLUME-SET = *CONTROL-VOLUME-SET

Subject to restrictions (see general operand description).

The file is stored in a control volume set of the SM pubset. S0-MIGRATION=*FORBIDDEN is set implicitly.

If the volume set does not match the requested file attributes, the command is rejected.

VOLUME-SET = <cat-id 1..4>

Subject to restrictions (see general operand description).

The file is stored in the specified volume set of the SM pubset. S0-MIGRATION=*FORBIDDEN is set implicitly. If the volume set does not match the requested file attributes, the command is rejected.

VOLUME = *STD / list-poss(255): <vsn 1..6>

VSN of a public disk or of a Net-Storage volume on which the file is to be created.

VOLUME = *STD

The system selects appropriate VSNs.

VOLUME = list-poss(255): <vsn 1..6>

Systems support can specify volume serial numbers explicitly.

Nonprivileged users can specify a VSN for a Net-Storage volume explicitly.

Nonprivileged users cannot specify a VSN explicitly for a public disk explicitly unless they have authorization to perform physical allocation of public storage space (see [“Privileged functions” on page 2-380](#)).

VSNs can only be specified for public disks of the pubset whose pubset ID matches the catalog ID of the file name.

If the specified disk is part of a volume set in an SM pubset, S0-MIGRATION and MIGRATE=*FORBIDDEN are set implicitly.

When a file is to be created on Net-Storage, the Net-Storage volume specified must be allocated to the catalog ID of the file name.

When the file is to be stored on a specific Net-Storage volume without a default name, the VSN must be specified. Otherwise the file is stored on the default Net-Storage volume or, if this does not exist, the system selects a Net-Storage volume

DEVICE-TYPE = *BY-VOLUME / <device>

Type of device to which the required public disk or Net-Storage volume is assigned.

Only device types known within the system are accepted. In interactive mode, the possible device types are displayed with DEVICE-TYPE=?.

The volume type NETSTOR must be specified for Net-Storage volumes.

If at least one volume ID is specified with VOLUME, each specification of a disk device type which is known to the system is handled like the STDDISK specification.

DEVICE-TYPE = *BY-VOLUME

The disk device type belonging to the public disk is determined by the system.

DEVICE-TYPE = <device>

Systems support can specify device types explicitly.

When files are created on public volumes, nonprivileged users cannot specify a device type explicitly unless they have authorization to perform physical allocation (see [“Privileged functions” on page 2-380](#)).

SPACE = *STD / *RELATIVE(...) / *ABSOLUTE(...)

Influences the storage space allocation for the file. The SPACE operand is always evaluated.

SPACE = *STD

For its primary and secondary allocations, the file is assigned the values defined by systems support.

SPACE = *RELATIVE(...)

With this operand, the user reserves storage space for the file (relative allocation). In doing so, he must take into account the following: the space ceiling defined in the user entry may be exceeded only if the user is authorized to do so (PUBLIC-SPACE-EXCESS=*YES, see the output of the SHOW-USER-ATTRIBUTES command).

In order to minimize system administration time, the following points should be noted when specifying primary and secondary allocations:

- the primary allocation should correspond to the expected size of the file to be created;
- for large files, the primary and secondary allocations should be selected as a multiple of the administration units packet or segment.

In relation to the allocation of storage space, the effect of the BUFFER-LENGTH operand in the ADD-FILE-LINK command must also be taken into account: when the primary and secondary allocations are being determined, the file's block length should be taken into account; for BUFFER-LENGTH=*STD(SIZE=n), where $n \geq 2$, the following applies:

File type	SPACE operand	
	Primary allocation	Secondary allocation
SAM	$\geq 2n$	$\geq n$
ISAM (K-ISAM and NK-ISAM)	$> n$	
PAM (chained I/O)	> 0	

Table 33: Minimum values for primary and secondary space allocations

PRIMARY-ALLOCATION = <integer 1..2147483647>

Number of PAM pages to be allocated initially.

The specified number of PAM pages will be reserved immediately. The number should correspond to the expected size of the file.

When storage space is actually allocated, the specified number of PAM pages is rounded up to a multiple of **k** if required. At this **k** is the number of PAM pages in the smallest allocation unit in each case:

Pubset/volume set format	Smallest allocation unit in Kbytes	Rounding factor k
K pubset/volume set	6	3
NK2(6K) pubset/volume set	6	3
NK2(8K) pubset/volume set	8	4
NK2(64K) pubset/volume set	64	32
NK4(8K) pubset/volume set	8	4
NK4(64K) pubset/volume set	64	32

Table 34: Smallest allocation unit based on pubset/volume set format

The pubset/volume set format and the smallest allocation unit can also be determined by the user from the output of the SHOW-MASTER-CATALOG-ENTRY command.

SECONDARY-ALLOCATION = *STD / <integer 0..32767>

The number of PAM pages by which the storage space should be extended, if this should become necessary. The default is STD, the default value for the system. In contrast to the primary allocation, the secondary allocation does not take effect as soon as the CREATE-FILE command is executed, but only if the reserved space proves insufficient when the file is being created or extended. The system will then automatically increase the storage space allocation for the file by the number of PAM pages specified in SECONDARY-ALLOCATION (field *S-ALLOC* in the output of the SHOW-FILE-ATTRIBUTES command). The value for secondary allocation is doubled after each successful extension. It is no longer doubled when the maximum value set in the system has been reached. When storage space is actually allocated, the specified number of PAM pages is rounded up to a multiple of **k** if required. The rounding factor **k** is dependent on the pubset/volume set format, i.e. on the formatting of the associated disks (see the PRIMARY-ALLOCATION operand).

A specification of SECONDARY-ALLOCATION=0 prevents dynamic expansion of the file.

SPACE = *ABSOLUTE(...)

This operand value is allowed only in combination with VOLUME operands.

Systems support staff can allocate storage space explicitly. In the case of files on Net-Storage, an absolute assignment of storage space is not permitted.

Nonprivileged users are allowed to allocate it only if they have authorization to perform physical allocation of public storage space (see “Privileged functions” on page 2-380). With this operand, the user reserves storage space for the file by an absolute allocation, i.e. he specifies the number of the PAM page on which the storage space allocation is to start. In doing so, he must take the following into account: An absolute allocation will only be made if it is possible to reserve the whole of the specified number of blocks (SIZE) on *one* single disk. If more than one disk was specified in the VOLUME operand, then only the first of the named disks will be used. If there isn’t enough free storage space on the disk, the CREATE-FILE command will be rejected; there will be no partial allocation. Since an absolute allocation always refers to a single volume, a separate CREATE-FILE or MODIFY-FILE-ATTRIBUTES command must be issued for each volume. The secondary allocation for the file (see the SECONDARY-ALLOCATION operand) will have the value zero, i.e. if the number of pages specified in SIZE proves insufficient at a later point, there will be no automatic extension of the storage space.

FIRST-PAGE = <integer 1..2147483647>

Block number of the PAM page at which storage space reservation on the public disk is to start. Since storage space is only reserved in units of **k** PAM pages, where **k** is the number of PAM pages per smallest allocation unit, the value of FIRST-PAGE = $k * n + 1$ ($n \geq 0$).

SIZE = <integer 3..2147483647>

Number of PAM pages to be reserved on the public disk. SIZE must be a multiple of **k**. The amount of storage space available on a disk depends on its type and its initialization; systems support should therefore be asked what maximum value SIZE may have.

MANAGEMENT-CLASS = *NONE / <composed-name 1..8>

Only for files in SM pubsets Specifies whether the HSMS backup, archiving and long-term archiving functions are controlled by a management class defined using HSMS. See the “HSMS” manual [18] for further details. Management class allocations are rejected for files in SF pubsets.

MANAGEMENT-CLASS = *NONE

The file attributes set with the SAVE and MIGRATE operands are evaluated for file backup and migration.

MANAGEMENT-CLASS = <composed-name 1..8>

Name of the management class defined using HSMS. The specified management class must exist, and the user ID of the file owner must have access authorization.

USER-INFORMATION = *NONE / <c-string 1..8 with-low>

Specifies whether user information is to be added to the catalog entry. This information is not evaluated by the system. Its content and meaning are defined solely by the user. The default is *NONE, i.e. no information is added.

ADM-INFORMATION = *NONE / <c-string 1..8 with-low>

This operand is available only to privileged users (TSOS privilege). Specifies whether a text is to be added to the catalog entry as information for systems support. This information is not evaluated by the system. Its content and meaning are defined solely by systems support. The default is *NONE, i.e. no information is added.

SUPPORT = *PRIVATE-DISK(...)

The file is to be set up on a private disk.

VOLUME = *ANY(...) / list-poss(255): <alphanum-name 1..6>

The VSN (volume serial number) of the disk on which the file is to be set up.

VOLUME = *ANY(...)

Indicates that any private disk should be mounted on the specified device type (no particular VSN is required).

The VSNs of the volumes which were requested are copied into the volume list in the catalog entry. If the user requests more than one volume to be mounted, the NUMBER-OF-DEVICES operand should be used to specify how many volumes are to be mounted at the same time.

NUMBER-OF-DEVICES = 1 / <integer 1..9>

Number of disks required.

VOLUME = list-poss(255): <alphanum-name 1..6>

A maximum of 255 different VSNs may be specified, i.e. for each CREATE-FILE command a maximum of 255 volumes may be requested. Each VSN may be specified only once. Starting with the volume on which the file starts, all the VSNs are copied into the volume list of the catalog entry.

For private disks, DMS attempts to reserve the whole of the storage space, requested by the SPACE operand, on the first of the private disks (see also the SPACE operand).

DEVICE-TYPE = *BY-VOLUME-CATALOG / <device>

The type of device to which the required disks are allocated. Only device types known within the system are accepted. In interactive mode, the possible device types are displayed with DEVICE-TYPE=?. If at least one volume ID is specified with VOLUME, each specification of a disk device type which is known to the system is handled like the STDDISK specification.

The permissible specifications for DEVICE-TYPE can also be found in the device table in [section "Device type table" on page 1-82](#) (device type column)). The devices available on the system can be listed with the SHOW-DEVICE-CONFIGURATION command.

DEVICE-TYPE = *BY-VOLUME-CATALOG

The device type is determined via the MAREN device substitution function if the MAREN subsystem is available.

SPACE = *STD / *RELATIVE(...) / *ABSOLUTE(...)

Influences the storage space allocation for the file. The SPACE operand is always evaluated.

Storage space allocation is described in the “Introductory Guide to DMS” [13].

SPACE = *STD

For its primary and secondary allocations, the file is assigned the values defined by the system parameters DMPRALL and DMSCALL.

SPACE = *RELATIVE(...)

With this operand, the user reserves storage space for the file (relative allocation).

In order to minimize system administration time, the following points should be noted when specifying primary and secondary allocations:

- the primary allocation should correspond to the expected size of the file to be created;
- for large files, the primary and secondary allocations should be selected as a multiple of the administration units packet or segment.

In relation to the allocation of storage space, the effect of the BUFFER-LENGTH operand in the ADD-FILE-LINK command must also be taken into account: when the primary and secondary allocations are being determined, the file’s block length should be taken into account; for BUFFER-LENGTH=*STD(SIZE=n), where $n \geq 2$, the following applies:

File type	SPACE operand	
	Primary allocation	Secondary allocation
SAM	$\geq 2n$	$\geq n$
ISAM (K-ISAM and NK-ISAM)	$> n$	
PAM (chained I/O)	> 0	

Table 35: Minimum values for primary and secondary space allocations

PRIMARY-ALLOCATION = <integer 1..2147483647>

Number of PAM pages to be allocated initially. The specified number of PAM pages will be reserved immediately. The number should correspond to the expected size of the file. The PRIMARY-ALLOCATION specification is rounded up to a multiple of 3, and the corresponding number of PAM pages are allocated on the private disk specified by the VOLUME operand. The system attempts to reserve the primary allocation on the first of the disks specified.

When large files are to be created, and when there are only a few segments free on each disk, specifying an appropriate primary allocation will initiate a disk swap. This then avoids excessive fragmentation of the file.

SECONDARY-ALLOCATION = *STD / <integer 0..32767>

The number of PAM pages by which the storage space should be extended, if this should become necessary. The default is STD, the default value for the system. In contrast to the primary allocation, the secondary allocation does not take effect as soon as the CREATE-FILE command is executed, but only if the reserved space proves insufficient when the file is being created or extended. The system will then automatically increase the storage space allocation for the file by the number of PAM pages specified in SECONDARY-ALLOCATION (field *S-ALLOC* in the output of the SHOW-FILE-ATTRIBUTES command). The value for secondary allocation is doubled after each successful extension. It is no longer doubled when the maximum value set in the system has been reached. If required, the defined number of PAM pages is rounded up to a multiple of 3 when allocating space.

A specification of SECONDARY-ALLOCATION=0 prevents dynamic expansion of the file.

SPACE = *ABSOLUTE(...)

Absolute allocation (only meaningful when a VSN is specified in the VOLUME operand).

With this operand, the user reserves storage space for the file by an absolute allocation, i.e. he specifies the number of the PAM page on which the storage space allocation is to start. In doing so, he must take the following into account: An absolute allocation will only be made if it is possible to reserve the whole of the specified number of blocks (SIZE) on *one* single disk. If more than one disk was specified in the VOLUME operand, then only the first of the named disks will be used. If there isn't enough free storage space on the disk, the CREATE-FILE command will be rejected; there will be no partial allocation. Since an absolute allocation always refers to a single volume, a separate CREATE-FILE or MODIFY-FILE-ATTRIBUTES command must be issued for each volume. The secondary allocation for the file (see the SECONDARY-ALLOCATION operand) will have the value zero, i.e. if the number of pages specified in SIZE proves insufficient at a later point, there will be no automatic extension of the storage space.

FIRST-PAGE = <integer 1..2147483647>

The block number of the PAM page on which the storage space reservation on the private disk is to start. Since storage space is only reserved in units of three PAM pages, FIRST-PAGE is limited by: $\text{FIRST-PAGE} = 3n + 1$ ($n \geq 0$). The PAM page on which storage space reservations can start on any disk will depend on how the disk is initialized.

SIZE = <integer 1..2147483647>

Specifies how many PAM pages are to be reserved on the disk; SIZE must be a multiple of three. The amount of storage space available on a disk depends on its type and its initialization; systems support should therefore be asked what maximum value SIZE may have.

DATA-SUPPORT = *SAME / *PARAMETERS(...)

Only for ISAM files: where the data section is stored. This operand is used if the data and index blocks for an ISAM file are to be stored on separate disks.

DATA-SUPPORT = *SAME

The data and index blocks of the ISAM file are not to be stored on separate disks.

DATA-SUPPORT = *PARAMETERS(...)

The data and index blocks of the ISAM file are to be stored on separate disks. The values specified in the operands VOLUME, DEVICE-TYPE, and SPACE are only applicable to the index section of the file.



Only K-ISAM files can be separated into index and blocks. The values specified for the operands DATA-VOLUME, DATA-DEVICE-TYPE, and DATA-SPACE are ignored for NK-ISAM files.

DATA-DEVICE-TYPE = <device>

For K-ISAM files with separate index and data sections: The device type to which the required disks are assigned. Only device types known within the system are accepted. In interactive mode, the possible device types are displayed with DEVICE-TYPE=?. If at least one volume ID is specified with DATA-VOLUME, each specification of a disk device type which is known to the system is handled like the STDDISK specification.

The disk type for the data section is named with DATA-DEVICE-TYPE (for the index section with DEVICE-TYPE). The permissible specifications for DEVICE-TYPE can also be found in the device table in [section "Device type table" on page 1-82](#) (device type column)). DATA-DEVICE-TYPE must be specified if no storage has yet been reserved for the file. Although NK-ISAM does not support separation of index and data sections, it is possible to specify DATA-DEVICE-TYPE (for compatibility with K-ISAM).

DATA-VOLUME = list-poss(255): <alphanum-name 1..6>

In conjunction with DATA-DEVICE-TYPE for ISAM files with separate index and data sections.

The VSN of the private disk on which the data section of the ISAM file is to be set up. A maximum of 255 VSNs may be specified, but each VSN may be specified only once.

Although NK-ISAM does not support separation of index and data sections, it is possible to specify DATA-VOLUME (for compatibility with K-ISAM). DATA-VOLUME specifies the VSN of the volume on which the data section of the ISAM file is to be stored; the VOLUME operand must be specified for the index section. The further details are analogous to those for DATA-DEVICE-TYPE.

DATA-SPACE = *RELATIVE(...) / *ABSOLUTE(...)

In conjunction with DATA-DEVICE-TYPE/DATA-VOLUME for the data section of ISAM files with separation of index and data sections Defines the storage space allocations for the data section of an ISAM file. In contrast to the SPACE operand, the user must explicitly specify the storage space allocation, since the system does not provide a default allocation (minimal specification: the primary allocation under SPACE=*RELATIVE).

The rules for specifying primary, secondary, and absolute allocations correspond to those for the SPACE operand, but relate to the volumes listed in DATA-VOLUME instead (see also the operands DATA-DEVICE, DATA-VOLUME). Although NK-ISAM does not support separation of index and data sections, it is possible to specify DATA-SPACE (for compatibility with K-ISAM).

DATA-SPACE = *RELATIVE(...)

With this operand, the user can reserve storage space for the data section of an ISAM file in the form of a relative allocation.

PRIMARY-ALLOCATION = <integer 1..2147483647>

Number of PAM pages to be allocated initially. The specified number of PAM pages will be reserved immediately. The number should correspond to the expected size of the file. The system will attempt to make the initial allocation on the first disk specified.

SECONDARY-ALLOCATION = *STD / <integer 0..32767>

The number of PAM pages for later file extensions.

The default value is STD, the system default value.

The number of PAM pages specified here is not immediately reserved. Only when the number of pages initially reserved is no longer sufficient will the system automatically increase the storage space allocated to the data section of the ISAM file by the number of PAM pages specified in the SECONDARY-ALLOCATION. This allocation is made dynamically, i.e. each time that additional storage space is required. If it is not already a multiple of 3, the extra space requested will be rounded to one. The value of the secondary allocation is copied into the catalog entry.

DATA-SPACE = *ABSOLUTE(...)

With this operand, the user reserves storage space for the data section of the file by an absolute allocation, i.e. he specifies the number of the physical block on the disk at which the storage space allocation is to start. In doing so, the following must

be taken into account: An absolute allocation will only be made if it is possible to reserve the whole of the specified number of PAM pages (SIZE) on **one** single disk. If more than one disk was specified in the DATA-VOLUME operand, then only the first of the named disks will be used.

FIRST-PAGE = <integer 1..2147483647>

PAM page number at which the absolute assignment begins. (The number specified must be a multiple of 3, plus 1; e.g. 1 / 4 / 7 / 10 etc.).

SIZE = <integer 1..2147483647>

The number of PAM pages to be reserved (this must be a multiple of 3).

SUPPORT = *TAPE(...)

The file is to be created on a tape. The required tape need not be mounted immediately by the operator.

VOLUME = *NO / *ANY(...) / list-poss(255): <alphanum-name 1..6>

Volume serial numbers of the tapes on which the file is to be created. A maximum of 255 volume serial numbers may be specified.

VOLUME = *NO

No VSNs are to be entered into the catalog yet.

VOLUME = *ANY(...)

The operator may mount any suitable tapes on the specified device type (no particular VSNs are required).

The VSNs of the volumes which were requested are copied into the volume list in the catalog entry. If the user requests more than one volume to be mounted, the NUMBER-OF-DEVICES operand should be used to specify how many volumes are to be mounted at the same time.

NUMBER-OF-DEVICES = 1 / <integer 1..9>

The number of tapes which is required.

VOLUME = list-poss(255): <alphanum-name 1..6>

A maximum of 255 different VSNs may be specified, i.e. for each CREATE-FILE command a maximum of 255 volumes may be requested. Each VSN may be specified only once.

DEVICE-TYPE = *BY-VOLUME-CATALOG / <device>

Device type to which the required tapes are assigned. The value to be specified is the device type or volume type of the tapes, from which the Device Management System determines the device type which is to be made available for tape processing. Only device types known within the system are accepted. In interactive mode, the possible device types are displayed with DEVICE-TYPE=?.

If TAPE is specified as the device type, magnetic tape devices that support a recording density of 1600 or 6250 bpi (bytes per inch) are assigned. Other permissible specifications can be found in section [“Volume types for magnetic tape devices” on page 1-84](#).

DEVICE-TYPE=WORK requests a work tape. Work tapes are identified by having no tape owner entry in the VOL1 label (contents of “Owner identifier” field: X'40'). Work tapes should be requested if they are required only for the duration of the processing, and are not to be archived. If DEVICE-TYPE=WORK is specified, the operands VOLUME and PREMOUNT-LIST will be ignored.

DEVICE-TYPE=WORK should not be specified for file sets, because the work tape which is allocated is always that which is immediately available.

Tape cartridges cannot be requested as work tapes.

DEVICE-TYPE = *BY-VOLUME-CATALOG

The device type is determined via the MAREN device substitution function if the MAREN subsystem is available.

PREMOUNT-LIST = *NONE / list-poss(255): <integer 0..255>

The tape sequence number of the tape to be mounted. Issues a MOUNT message on the console requesting the mounting of tapes or suppresses the request.

PREMOUNT-LIST = *NONE

The first volume specified in the VOLUME operand will be requested.

PREMOUNT-LIST = list-poss(255): <integer 0..255>

Issues a MOUNT message on the console requesting the mounting of tapes or suppresses the request. The tape sequence numbers specified here relate to the VSNs listed in the VOLUME operand. The volumes specified in this way are to be mounted by the operator (MOUNT request on the console). Multiple tape sequence numbers must be listed in ascending order (n,n+1,...).

Specifying PREMOUNT-LIST=0 has the effect that no tapes will be requested. Otherwise, it is only on opening the file that the first volume is requested.

SUPPORT = *NONE

A catalog entry will be created, but no storage space will be reserved yet for the file.

PROTECTION = *STD / *PARAMETERS(...)

The protection attributes for the file.

PROTECTION = *STD

The values supplied by default protection are employed for the protection attributes. The AUDIT and FREE-FOR-DELETION protection attributes are always set to *NONE.

If default protection is not active, the system default values for the operands of the *PARAMETERS structure are set.

PROTECTION-ATTR= Protection attribute	*FROM-FILE	*STD	*BY-DEF-PROT-OR-STD
		(System default values)	
			Default prot. not active Default protection active
ACCESS	value from reference file	WRITE	
USER-ACCESS		OWNER-ONLY	
BASIC-ACL		NONE	
DESTROY-BY-DELETE		NO	
PASSWORD		NONE	
SPACE-RELEASE-LOCK		NO	
READ-PASSWORD	NONE		
WRITE-PASSWORD			
EXEC-PASSWORD			
FREE-FOR-DELETION			
AUDIT			

Table 36: Effects of PROTECTION-ATTR on protection attributes in the case of CREATE-FILE

PROTECTION = *PARAMETERS(...)

This operand allows the following protection attributes to be specified for the file. If the default value *NONE or *NO is specified for an attribute, the corresponding protection function is not activated. In conjunction with the ACCESS, USER-ACCESS, BASIC-ACL, GUARDS and FREE-FOR-DELETION protection attributes and with password protection, only default protection attributes (see table 36) may be specified for temporary files. Since only the creating job can access temporary JVs, no protection is required against foreign access.

When the file is accessed, the highest activated protection level applies. The following table shows the method used for access control, the protection attributes, and the job variable protection hierarchy (protection levels):

Access control	Protection attribute	Prot. level
Standard access control	ACCESS and USER-ACCESS	0
Basic access control list	BASIC-ACL	1
Access control via guards	GUARDS using the software product SECOS	2

Table 37: Hierarchy of access control methods

Passwords are evaluated independently, without regard to the implemented protection level.

If the file's date of release for deletion has been reached (see the FREE-FOR-DELETION operand), the file owner can delete the file regardless of its access protection.

PROTECTION-ATTR = *BY-DEF-PROT-OR-STD / *STD / *FROM-FILE(...)

Specifies from where the protection attributes with the value *BY-PROTECTION-ATTR or *BY-PROT-ATTR-OR-NONE are to be obtained. For the assignments, see [table "Effects of PROTECTION-ATTR on protection attributes in the case of CREATE-FILE" on page 2-407](#).

PROTECTION-ATTR = *BY-DEF-PROT-OR-STD

The values supplied by default protection are entered for the operands with *BY-PROTECTION-ATTR or *BY-PROT-ATTR-OR-NONE. If default protection is not active, system default values or *NONE will be assigned to the relevant operands.

PROTECTION-ATTR = *STD

System default values are set for operands with *BY-PROTECTION-ATTR. *NONE is set for operands with *BY-PROT-ATTR-OR-NONE.

PROTECTION-ATTR = *FROM-FILE(...)

Operands for which *BY-PROTECTION-ATTR is specified, receive the value that they have in the catalog entry of the file named next. *NONE is set for operands with *BY-PROT-ATTR-OR-NONE.

If the protection attributes are to be copied for a temporary file, either the operands ACCESS=*WRITE, USER-ACCESS=*OWNER-ONLY, BASIC-ACL=*NONE and GUARDS=*NONE must be specified explicitly or the specified file must have been cataloged with these attributes. ACCESS and USER-ACCESS can have any value in the case of temporary tape files.

FILE-NAME = <filename 1..54 without-gen>

Reference file from which the protection attributes with *BY-PROTECTION-ATTR are to be copied. The file has to be on the same pubset as the new file currently being created. If the file does not exist or if it is not accessible (e.g. with SHOW-FILE-ATTRIBUTES), then the command is rejected.

ACCESS = *BY-PROTECTION-ATTR / *WRITE / *READ

The ACCESS operand enables a file to be protected against overwriting; it specifies whether write access to the file is permitted (implicitly allowing read access) or only read access. This protection attribute is part of the standard access control mechanism and is only evaluated if no higher protection level has been activated (see [table “Hierarchy of access control methods” on page 2-407](#)).

Tape files: When the file is first opened, DMS copies the ACCESS type identifier into the HDR3 label. During subsequent accesses to the file, the file owner can bypass the checks on access type by using the command ADD-FILE-LINK... PROTECTION-LEVEL=*LOW.

ACCESS = *BY-PROTECTION-ATTR

The value defined for the file specified in the PROTECTION-ATTR operand is used.

ACCESS = *WRITE

All types of access to the file are permitted.

Tape files: HDR3 label: access type identifier = 0

ACCESS = *READ

The file may only be accessed to read it.

Temporary files: Write access cannot be prevented, ACCESS=*READ will be rejected.

Tape files: HDR3 label: access type identifier = 1

USER-ACCESS = *BY-PROTECTION-ATTR / *OWNER-ONLY / *ALL-USERS / *SPECIAL

Specifies whether the file may be accessed by user IDs other than that of its owner or co-owners. The access type permitted in this case is governed by the ACCESS protection attribute (see the ACCESS operand).

This protection attribute is part of the standard access control mechanism. It is evaluated for users without HARDWARE-MAINTENANCE privileges only if no higher access protection is defined for the file (see [table “Hierarchy of access control methods” on page 2-407](#)). *Tape files:* When the file is first opened, DMS copies the SHARE indicator into the HDR1 label (“access indicator”).

USER-ACCESS = *BY-PROTECTION-ATTR

The value defined for the file specified in the PROTECTION-ATTR operand is used.

USER-ACCESS = *OWNER-ONLY

The file is not shareable, which means that only the file owner and any co-owners have access to it.

Tape files: HDR1 label: access indicator = 1

USER-ACCESS = *ALL-USERS

The file is shareable, which means that any user ID can access it.

Temporary files: USER-ACCESS=ALL-USERS is not permitted.

Tape files: HDR1 label access indicator = (X'40') ◡

USER-ACCESS = *SPECIAL

User IDs with HARDWARE-MAINTENANCE privileges (online maintenance) may access the file. USER-ACCESS=*ALL-USERS applies implicitly, which means that the file can be accessed by all user IDs, including the maintenance IDs. If accesses under a maintenance ID are to be permitted, SPECIAL must be set even if BASIC-ACL or GUARDS is used for access control.

BASIC-ACL = *BY-PROTECTION-ATTR / *NONE / *STD / *PARAMETERS(...) / *GROUP-X / *GROUP-RX / *GROUP-WRX / *ALL-X / *ALL-RX / *ALL-WRX

Specifies whether a BASIC-ACL entry should be created for the file, and hence whether this entry is to be used for access control. A BASIC-ACL entry will only be evaluated if no higher protection level has been activated for the file (see also [table "Hierarchy of access control methods" on page 2-407](#)). The following operand values correspond to common BASIC-ACL settings:

Operand value	BASIC-ACL protection								
	OWNER			GROUP			OTHERS		
	R	W	X	R	W	X	R	W	X
*STD	Y	Y	Y	N	N	N	N	N	N
*GROUP-X	Y	Y	Y	N	N	Y	N	N	N
*GROUP-RX	Y	Y	Y	Y	N	Y	N	N	N
*GROUP-WRX	Y	Y	Y	Y	Y	Y	N	N	N
*ALL-X	Y	Y	Y	N	N	Y	N	N	Y
*ALL-RX	Y	Y	Y	Y	N	Y	Y	N	Y
*ALL-WRX	Y	Y	Y	Y	Y	Y	Y	Y	Y

Y: access granted N: access denied

Table 38: Meaning of the operand values for common BASIC-ACL settings

BASIC-ACL = *BY-PROTECTION-ATTR

The value defined for the file specified in the PROTECTION-ATTR operand is used.

BASIC-ACL = *NONE

No BASIC ACL entry will be created for the file. Access will be controlled as defined in the USER-ACCESS and ACCESS entries.

BASIC-ACL = *PARAMETERS(...)

A BASIC-ACL entry is created for the file, and the BASIC-ACL (basic access control list) is used to control access, provided no higher protection level has been activated (see [table 37 on page 2-407](#)). Read, write and execute authorizations must be explicitly set for each user ID which is to have access rights, otherwise no access right will be granted. Those with access rights are:

- OWNER, i.e. the user ID of the file owner and the user IDs of the co-owners and of systems support (see “Privileged functions”).
- GROUP, i.e. all user IDs which are in the same group as the file owner (excluding the owner, the co-owners and systems support). Definition of user groups is possible only when the software product SECOS is used. However, to allow for possible later installation of SECOS, GROUP should be given the same access rights as for OTHERS.
- OTHERS, i.e. all user IDs which belong to a different group from the file owner (excluding the owner, the co-owners and systems support).

OWNER = *NO-ACCESS / *PARAMETERS(...)

Specifies what access rights are to be given to the owner and the co-owners. The default setting is *NO-ACCESS, i.e. there is no read, write or execute permission.

OWNER = *PARAMETERS(...)

The access rights will be as specified:

READ = *NO / *YES

Specifies whether read authorization is set.

The default setting is *NO, i.e. there is no read permission.

WRITE = *NO / *YES

Specifies whether write authorization is specified.

The default setting is *NO, i.e. there is no write permission. Write permission does not imply read permission.

EXEC = *NO / *YES

Indicates whether read access is permitted or not.

The default setting is *NO, i.e. there is no execute permission.

GROUP = *NO-ACCESS / *PARAMETERS(...)

Specifies which access rights are to be set for all user IDs from the group of the owner. The default setting is *NO-ACCESS, i.e. there is no read, write or execute permission.

GROUP = *PARAMETERS(...)

Access rights are to be set as specified:

READ = *NO / *YES

Specifies whether read authorization is set.

The default setting is *NO, i.e. there is no read permission.

WRITE = *NO / *YES

Specifies whether write authorization is specified.

The default setting is *NO, i.e. there is no write permission. Write permission does not imply read permission.

EXEC = *NO / *YES

Indicates whether read access is permitted or not.

The default setting is *NO, i.e. there is no execute permission.

OTHERS = *NO-ACCESS / *PARAMETERS(...)

Specifies what access rights are to be given to all other user IDs. Unless the software product SECOS is in use, the access authorizations defined for GROUP and OTHERS should be the same. The default setting is *NO-ACCESS, i.e. there is no read, write or execute permission.

OTHERS = *PARAMETERS(...)

Access rights are to be set as specified:

READ = *NO / *YES

Specifies whether read authorization is set. The default setting is *NO, i.e. there is no read permission.

WRITE = *NO / *YES

Specifies whether write authorization is specified.

The default setting is *NO, i.e. there is no write permission. Write permission does not imply read permission.

EXEC = *NO / *YES

Indicates whether read access is permitted or not.

The default setting is *NO, i.e. there is no execute permission.

GUARDS = *BY-PROTECTION-ATTR / *NONE / *PARAMETERS(...)

Specifies whether access to the file is controlled by GUARDS

GUARDS = *BY-PROTECTION-ATTR

The value defined for the file specified in the PROTECTION-ATTR operand is used.

GUARDS = *NONE

Access to the JV is not to be controlled via GUARDS.

GUARDS = *PARAMETERS(...)

Access to the JV is to be controlled via GUARDS.

Access to the file is controlled via a guard, i.e. a specific object identifying all the conditions subject to which access will be granted: such as date, time and user ID. The GUARDS function unit of the chargeable software product SECOS (see the "SECOS" manual [35]) must be installed in order to create and maintain a guard. Each guard is uniquely identified by its name. The guard name resembles a file name: it can contain a user ID and consists of a name part which is up to 8 characters in length. If no user ID is specified explicitly, the user's own ID is added implicitly. Each access mode can be controlled by a separate guard. If no guard is assigned for an access mode (*NONE), access control will refuse any corresponding access (e.g. WRITE=*NONE prevents all write access). Specifying GUARDS=*PARAMETERS defines access control using GUARDS with the default value *NONE for all access modes, which means that the file cannot be read, updated or executed. The GUARDS subsystem is not required in order to define access control via GUARDS. The appropriate checks by GUARDS are not performed until the the time file access takes place: If a guard has been defined but is not available, all access of the type controlled by that guard is prohibited. No access at all is possible if the GUARDS subsystem is not available at the time of access.

READ = *NONE / <filename 1..18 without-cat-gen-vers>

Name of a guard controlling read access (up to 8 characters if no user ID is specified). The default value is *NONE, i.e. no read access is granted.

WRITE = *NONE / <filename 1..18 without-cat-gen-vers>

Name of a guard controlling write access (up to 8 characters if no user ID is specified). The default value is *NONE, i.e. no write access is granted.

EXEC = *NONE / <filename 1..18 without-cat-gen-vers>

Name of a guard for execute protection (up to 8 characters when no user ID is specified). The default setting is *NONE, i.e. the file cannot be executed.

WRITE-PASSWORD = *BY-PROT-ATTR-OR-NONE / *NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647> / *SECRET

Write or read password for the JV to be modified.

The password defined here must be specified in the ADD-PASSWORD command to enable write access to the file.

The WRITE-PASSWORD operand has the following special characteristics:

- The password entered is not logged.
- The input field is automatically blanked out in the guided dialog.
- In unguided dialog and foreground procedures, the entry *SECRET or ^, SDF provides a blanked out input field for inputting the password .

Temporary files: cannot be password-protected.

Tape files: the password protection is noted in the HDR3 label.

WRITE-PASSWORD = *BY-PROT-ATTR-OR-NONE

Allocation of a write password is independent of the value of the PROTECTION-ATTR operand.

If PROTECTION-ATTR=*BY-DEF-PROT-OR-STD is set, the values supplied by default protection or - if default protection is not activated - *NONE is entered.

With PROTECTION-ATTR=*STD / *FROM-FILE(), *NONE is entered, i.e. no write protection password is used.

WRITE-PASSWORD = *NONE

No write protection password is specified.

**WRITE-PASSWORD = <c-string 1..4> / <x-string 1..8> /
<integer -2147483648..2147483647>**

Defines a password required for write accesses.

**READ-PASSWORD = *BY-PROT-ATTR-OR-NONE / *NONE / <c-string 1..4> /
<x-string 1..8> / <integer -2147483648..2147483647> / *SECRET**

Password for protection against unauthorized reading.

The password defined here must be specified in the ADD-PASSWORD command to enable read access to the file.

The READ-PASSWORD operand has the following special characteristics:

- The password entered is not logged.
- The input field is automatically blanked out in the guided dialog.
- In unguided dialog and foreground procedures, the entry *SECRET or ^, SDF provides a blanked out input field for inputting the password .

Temporary files: cannot be password-protected.

Tape files: the password protection is noted in the HDR3 label.

READ-PASSWORD = *BY-PROT-ATTR-OR-NONE

Allocation of a read password is independent of the value of the PROTECTION-ATTR operand.

If PROTECTION-ATTR=*BY-DEF-PROT-OR-STD is set, the values supplied by default protection or - if default protection is not activated - *NONE is entered.

With PROTECTION-ATTR=*STD / *FROM-FILE(), *NONE is entered, i.e. no read protection password is used.

READ-PASSWORD = *NONE

No read protection password is specified.

**READ-PASSWORD = <c-string 1..4> / <x-string 1..8> /
<integer -2147483648..2147483647>**

Defines a password required for read accesses.

A source program which is protected by a read password cannot be compiled.

EXEC-PASSWORD = *BY-PROT-ATTR-OR-NONE / *NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647> / *SECRET

The password used to protect against unauthorized execution of the file.

This type of password is specified for procedures and load modules. The password defined here must be specified in the ADD-PASSWORD command before the procedure or program can be run.

The operand has the following special characteristics:

- The password entered is not logged.
- The input field is automatically blanked out in the guided dialog.
- In unguided dialog and foreground procedures, the entry *SECRET or ^, SDF provides a blanked out input field for inputting the password .

EXEC-PASSWORD enables “execute passwords” to be defined.

The “execute” protection applies to a program or a procedure/ENTER file call, i.e. the file specified in the used command under FILE-NAME must contain an executable program or a procedure.

Temporary files: cannot be password-protected.

Tape files: the password protection is noted in the HDR3 label.

EXEC-PASSWORD = *BY-PROT-ATTR-OR-NONE

Assignment of an execution protection password depends on the value of the PROTECTION-ATTR operand.

If PROTECTION-ATTR=*BY-DEF-PROT-OR-STD is set, the values supplied by default protection or - if default protection is not activated - *NONE is entered.

With PROTECTION-ATTR=*STD / *FROM-FILE(), *NONE is entered, i.e. no execution protection password is used.

EXEC-PASSWORD = *NONE

No execute protection password is specified.

EXEC-PASSWORD = <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>

Defines a password which is required to call a program/procedure file.

If EXEC-PASSWORD=X'00000000' is specified, it will be ignored.

DESTROY-BY-DELETE = *BY-PROTECTION-ATTR / *NO / *YES

To increase the level of protection of data, the user can specify in the catalog entry that any data no longer required is overwritten by X'00' (binary zero). For disk files, DESTROY-BY-DELETE affects deletion, file migration and storage space release operations (see the MODIFY-FILE-ATTRIBUTES and DELETE-FILE commands); for tape files, any data remaining on the tape when EOF or EOVS processing is carried out will be overwritten (see also the ADD-FILE-LINK command, DESTROY-OLD-CONTENTS operand).

DESTROY-BY-DELETE = *BY-PROTECTION-ATTR

The value defined for the file specified in the PROTECTION-ATTR operand is used.

DESTROY-BY-DELETE = *NO

If this operand is set to *NO, the action specified in the DELETE-FILE command (OPTION operand) will be carried out.

Disk files: the storage space released will be left unchanged if the operand OPTION=DESTROY-ALL is not specified in the DELETE-FILE command.

Tape files: Any data which follows on the tape will not be overwritten if DESTROY-OLD-CONTENTS=*YES is not specified in the ADD-FILE-LINK command for the current processing run.

DESTROY-BY-DELETE = *YES

The setting *YES for this operand will be effective even if a different specification has been made in the OPTION operand of the DELETE-FILE command.

Disk files: when the storage space is released or deleted, it will be automatically overwritten with binary zeros (X'00').

Tape files: the contents of the tape following the end-of-file will be overwritten with binary zeros (X'00'). It is not necessary to explicitly set the ADD-FILE-LINK command to delete residual data for the current processing run (i.e. the operand DESTROY-OLD-CONTENTS=*YES).

AUDIT = *STD / *NONE / *SUCCESS / *FAILURE / *ALL

This operand specifies whether accesses to the file are to be monitored. This monitoring may also be made dependent on the result of the access (see the operand values *SUCCESS and *FAILURE). Systems support staff (TSOS privilege) can use this function unrestrictedly. Nonprivileged users require the appropriate authorization in the user entry of the pubset in which the file is to be created (see the *FILE-AUDIT* output field of the SHOW-USER-ATTRIBUTES command). Without this authorization the command will be rejected if auditing is requested.

It is possible to monitor the following DMS actions:

- the creation, amendment (protection attributes), reading and deletion of a catalog entry
- the opening and closing of files
- the renaming, importing and exporting of files

Monitoring is performed by System Exit Routines or, if the software product is being used, by the SAT function unit. In order to use System Exit Routines for monitoring, they must be activated (this is the responsibility of the system support staff, user ID TSOS). In order to use SAT for monitoring, SAT logging must have been started, and events for the FILE to be monitored must not be excluded from the logging procedure (this is the responsibility of the security officer, under the user ID SYSPRIV). The user cannot alter any explicit settings of SAT made by the security officer. The SAT log files can be analyzed by the SAT file administrator, under the user ID SYSAUDIT.

AUDIT = *STD

The value is currently equivalent to *NONE.

AUDIT = *NONE

No monitoring.

AUDIT = *SUCCESS

All successful DMS operations on the file will be monitored.

AUDIT = *FAILURE

All unsuccessful DMS operations on the file will be monitored.

AUDIT = *ALL

All DMS operations on the file will be monitored.

SPACE-RELEASE-LOCK = *BY-PROTECTION-ATTR / *NO / *YES

Specifies whether the releasing storage space is allowed.

SPACE-RELEASE-LOCK = *BY-PROTECTION-ATTR

The value defined for the file specified in the PROTECTION-ATTR operand is used.

SPACE-RELEASE-LOCK = *NO

Space is not released.

SPACE-RELEASE-LOCK = *YES

Space is released.

FREE-FOR-DELETION = *NONE / <date> / <integer 0..99999>

Specifies the date from which the owner is allowed to delete the file regardless of the ACCESS, BASIC-ACL, GUARDS, EXPIRATION-DATE protection attributes and regardless of the password (release date for deletion).

FREE-FOR-DELETION = *NONE

No free-for-deletion date is defined. The protection attributes are taken into account with regard to file deletion.

FREE-FOR-DELETION = <date>

Once the specified date is reached, the file can be deleted regardless of its protection attributes listed above. The date is specified in the form [yy]yy-mm-dd. 20 is automatically prefixed to two-digit year specifications < 60, 19 to two-digit year specifications ≥ 60.

FREE-FOR-DELETION = <integer 0..99999>

After the specified number of days, the file can be deleted regardless of its protection attributes named above.

The deletion release date must be specified in the form +n.

SAVE = *STD / *NO / *PARAMETERS(...)

Specifies how the file is to be treated when backups are performed with the ARCHIVE or HSMS utility if no management class has been assigned.

SAVE = *STD

The default values set under PARAMETERS will be applied if no management class has been assigned.

SAVE = *NO

This specification is equivalent to BACKUP-CLASS=*E and SAVED-PAGES=*COMPLETE-FILE under *PARAMETERS (q.v.).

SAVE = *PARAMETERS(...)

Specifies the frequency and type of backup. These details are taken into account when regular backups are carried out (using the ARCHIVE or HSMS utility).

BACKUP-CLASS = *STD / *A / *B / *C / *D / *E

Only for disk files. Backup frequency, where *A represents the most frequent backups. BACKUP relates to the automatic backup of files using either ARCHIVE or HSMS, and specifies which files are to be backed up during each backup run.

BACKUP-CLASS = *STD

The value defined by the system parameter BACKUP.

BACKUP-CLASS = *A

The most frequent backup. Files with this setting will be backed up by every backup run.

BACKUP-CLASS = *B

Files with this setting will be backed up every time a backup run is performed for files with BACKUP-CLASS=*B or *C or *D.

BACKUP-CLASS = *C

Files with this setting will be backed up every time a backup run is performed for files with BACKUP-CLASS=*C or *D.

BACKUP-CLASS = *D

The least frequent backup. Files with this setting will only be backed up when a backup run is performed for files with BACKUP-CLASS=*D.

BACKUP-CLASS = *E

Not backed up by ARCHIVE or HSMS. This value is the only possible value for temporary files (and is the default value). It is also appropriate for permanent files if, for example, work files are involved.

SAVED-PAGES = *COMPLETE-FILE / *MODIFIED-PAGES

Only for disk files: like BACKUP-CLASS, refers to the backup with ARCHIVE or HSMS and defines whether the entire file is to be backed up completely during automatic backup runs, or whether only those blocks that have changed since the last backup are to be saved.

SAVED-PAGES = *COMPLETE-FILE

The complete file is backed up.

SAVED-PAGES = *MODIFIED-PAGES

Incremental saving: only the modified blocks are saved. This is an appropriate specification for large files.

MIGRATE = *STD / *ALLOWED / *INHIBITED / *FORBIDDEN

This specification is relevant only for files on public disks. The operand is evaluated by the software product HSMS (Hierarchical Storage Management System). It allows a user to specify whether or not files that have not been accessed for a long time (by the user) may be migrated to a storage medium with slower access. The files will be migrated from the online processing level S0 to the background level S1, which is available online, or to background level S2 (e.g. tape storage) which is available offline (for further details see the "HSMS" manual [18]).

MIGRATE = *STD

Depends on the type of file or its storage location, the following default values apply:

*ALLOWED	for permanent files
*INHIBITED	for temporary files
*FORBIDDEN	for files created directly on the disk of an SM pubset (VOLUME operand)

MIGRATE = *ALLOWED

If HSMS is being used, the file may be migrated from the processing level (S0) to the background level (S1), or to the archive level (S2).

MIGRATE = *INHIBITED

The file may only be migrated for a brief period, e.g. for reorganization purposes (migration lock).

MIGRATE = *FORBIDDEN

This specification is allowed only in conjunction with authorization to perform physical allocation of public storage space (see "Privileged functions" under "Function" above). The file must not be migrated, typically because it is intended to stay in its current physical storage location (intensified migration lock).

CODED-CHARACTER-SET = *USER-DEFAULT / *NONE / <name 1..8>

This operand is ignored with regard to files on private disks. Determines the code of the file. This defines how the characters of a national character set are to be stored in binary form. The specified character set has an effect on the representation of characters on the screen, the collating sequence, etc. (see the "XHCS" manual [51]).

CODED-CHARACTER-SET = *USER-DEFAULT

The code is taken over from the file owner's user ID if a code other than EDF03IRV is entered there, otherwise *NONE applies.

CODED-CHARACTER-SET = *NONE

No coded character set is specified for the file.

CODED-CHARACTER-SET = <name 1..8>

Specifies the coded character set for the file.

SUPPRESS-ERRORS = *NONE / *FILE-EXISTING

Determines whether the existence of the specified file is handled as an error.

SUPPRESS-ERRORS = *NONE

If the specified file already exists, the command is rejected, and error handling is initiated (spin-off mechanism resp. The error triggers the spin-off mechanism resp. the SDF-P error handling).

SUPPRESS-ERRORS = *FILE-EXISTING

If the specified file already exists, the command has no effect on the file and its catalog entry. No error is reported.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed
2	0	DMS051E	Inconsistency in private disk pool or operand error detected
2	0	DMS0546	Catalog entry for specified file has reached maximum size
2	0	DMS054A	Insufficient disk space or access to disk not possible
	1	CMD0202	Syntax or semantic error in command
	32	DMS0584	A state that does not allow the function to continue was reported during processing
	32	DMS05C7	Unexpected internal error in DMS
	64	CMD0216	Privilege errors
	64	DMS0501	Requested catalog not available
	64	DMS0512	Requested catalog not found
	64	DMS051B	Requested user ID not in pubset Guaranteed messages: DMS051B, DMS0681
	64	DMS051C	User not authorized to access pubset Guaranteed messages: DMS051C, DMS0681
	64	DMS053E	File already cataloged on private volume
	64	DMS0540	Volume does not support requested file attributes
	64	DMS0557	Invalid volume specification
	64	DMS057A	Invalid combination of file attributes and storage class
	64	DMS057B	Invalid operand for migrated file
	64	DMS057C	Processing not possible due to HSMS error
	64	DMS057E	File has been migrated, and HSMS is not available
	64	DMS057F	Renaming of migrated file not permitted
	64	DMS0585	Error detected when processing catalog or multiprocessor system
	64	DMS0586	It is not possible to access or reserve a volume at present
	64	DMS0587	Use of the specified command has been restricted by the system administrator
	64	DMS0588	The disk storage assignment could not be executed
	64	DMS05B5	Guard not available
	64	DMS05BD	Invalid combination of file and volume set attributes
	64	DMS05FC	Specified user ID not in HOME pubset
	64	DMS060D	Invalid file name for reference file (PROTECTION-ATTR)
	64	DMS0613	Unknown management class
	64	DMS0618	Unknown storage class
	64	DMS061A	Storage class catalog could not be read
	64	DMS0640	Access to Net-Storage is rejected by the ONETSTOR subsystem because of communication problems with the net client
	64	DMS0641	File already exists on Net-Storage
	64	DMS0642	Large files not permitted on the specified pubset
	64	DMS0643	Net client reports access error
	64	DMS0644	Net client reports internal error
	64	DMS0645	File does not exist on Net-Storage
	64	DMS0647	Specified file type does not match the catalog entry

(Part 1 of 2)

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	64	DMS0648	Specification of file type, device and volume are not compatible
	64	DMS0649	Net server reports POSIX-ACL error
	64	DMS064A	Net client reports that access to files is forbidden on the Net-Storage volume
	64	DMS064B	Access to node files not supported by the net client
	64	DMS0652	Absolute storage space request not permitted on Net-Storage
	64	DMS0683	File already exists Guaranteed message: DMS0683
	64	DMS06FF	BCAM connection severed
	128	DMS0506	Function not executed due to change in master
	130	DMS0524	System address space exhausted
	130	DMS053C	No space in pubset catalog file
	130	DMS0582	File is currently locked or being used and cannot be processed
	130	DMS0585	Error detected when processing catalog or multiprocessor system Guaranteed message: DMS053C
	130	DMS0586	It is not possible to access or reserve a volume at present
	130	DMS0588	The disk storage assignment could not be executed
	130	DMS0594	Not enough virtual memory available
	130	DMS05C8	Maximum permitted number of files reached

(Part 2 of 2)

Examples

Example 1: Creating a catalog entry

```

/create-file file-name=max.file.1
/show-file-attr file-name=max.file.1,inf=*all-attributes
%0000000003 :20SG:$USER1.MAX.FILE.1
% ----- HISTORY -----
% CRE-DATE = NONE ACC-DATE = NONE CHANG-DATE = NONE
% CRE-TIME = NONE ACC-TIME = NONE CHANG-TIME = NONE
% ACC-COUNT = 0 S-ALLO-NUM = 0
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = NONE
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = NONE
% SP-REL-LOCK= NO ENCRYPTION = *NONE
% ----- BACKUP -----
% BACK-CLASS = A SAVED-PAG = COMPL-FILE VERSION = 0
% MIGRATE = ALLOWED
% ----- ORGANIZATION -----
% FILE-STRUC = NONE BUF-LEN = NONE BLK-CONTR = NONE
% IO(USAGE) = READ-WRITE IO(PERF) = STD DISK-WRITE = IMMEDIATE
% REC-FORM = NONE REC-SIZE = 0
% AVAIL = *STD
% WORK-FILE = *NO F-PREFORM = *K SO-MIGR = *ALLOWED
% ----- ALLOCATION -----
% SUPPORT = PUB S-ALLOC = 9 HIGH-US-PA = 0
% EXTENTS VOLUME DEVICE-TYPE EXTENTS VOLUME DEVICE-TYPE
% 1 GVS2.3 D3435
% NUM-OF-EXT = 1
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 3 REL= 3 PAGES

```

The output of the SHOW-FILE-ATTRIBUTES command shows the catalog entry that was generated for the file *MAX.FILE.1* using CREATE-FILE. Since no specification was made for the SUPPORT operand, storage space on public disk was reserved for the file.

Example 2: Creating a catalog entry for a tape file

```

/create-file file-name=max.tape-file.1,
             support=*tape(vol=m2326k,dev-type=tape-c4)
/show-file-attr file-name=max.tape-file.1,inf=*all-attributes
%           :20SG:$USER1.MAX.TAPE-FILE.1
% ----- HISTORY -----
% CRE-DATE   = NONE      ACC-DATE   = NONE      CHANG-DATE = NONE
% CRE-TIME   = NONE      ACC-TIME   = NONE      CHANG-TIME = NONE
% ACC-COUNT  = 0         S-ALLO-NUM = 0
% ----- SECURITY -----
% READ-PASS  = NONE      WRITE-PASS = NONE      EXEC-PASS  = NONE
% USER-ACC   = ALL-USERS ACCESS      = WRITE
% AUDIT      = NONE      FREE-DEL-D = *NONE     EXPIR-DATE = NONE
% DESTROY    = NO        FREE-DEL-T = *NONE     EXPIR-TIME = NONE
% ----- BACKUP -----
% BACK-CLASS = A         SAVED-PAG  = COMPL-FILE VERSION   = 0
% ----- ORGANIZATION -----
% FILE-STRUC = NONE      BUF-LEN    = NONE      BLK-CONTR = NONE
% REC-FORM   = NONE      REC-SIZE   = 0
% CODE       = NONE      LABEL        = NONE      FILE-SEQ  = NONE
% BLK-OFFSET = 0
% ----- ALLOCATION -----
% SUPPORT    = PVT
% EXTENTS    VOLUME      DEVICE-TYPE
% NONE       NONE        NONE
%:20SG: TAPE :          1 FILE

```

The output of the SHOW-FILE-ATTRIBUTES command shows the catalog entry that was generated with CREATE-FILE for the tape file *MAX.TAPE-FILE.1*, which is to be stored on the tape cartridge *M2326K* (volume type *TAPE-C4*). No reserved storage space is shown for the tape file, since only the catalog entry is saved in the file catalog of the pubset *20SG*.

Example 3: Creating a catalog entry for a file on private disk

```

/CREATE-FILE FILE-NAME=max.disk-file.1,support=*priv(vol=work01,
dev-type=d3435)
/SHOW-FILE-ATTR FILE-NAME=max.disk-file.1,inf=*all-attr
% ----- HISTORY -----
% CRE-DATE = NONE ACC-DATE = NONE CHANG-DATE = NONE
% CRE-TIME = NONE ACC-TIME = NONE CHANG-TIME = NONE
% ACC-COUNT = 0 S-ALLO-NUM = 0
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = NONE
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = NONE
% SP-REL-LOCK= NO ENCRYPTION = *NONE
% ----- BACKUP -----
% BACK-CLASS = A SAVED-PAG = COMPL-FILE VERSION = 0
% MIGRATE = ALLOWED
% ----- ORGANIZATION -----
% FILE-STRUC = NONE BUF-LEN = NONE BLK-CONTR = NONE
% IO(USAGE) = READ-WRITE IO(PERF) = STD DISK-WRITE = IMMEDIATE
% REC-FORM = NONE REC-SIZE = 0
% AVAIL = *STD
% ----- ALLOCATION -----
% SUPPORT = PVT S-ALLOC = 9 HIGH-US-PA = 0
% EXTENTS VOLUME DEVICE-TYPE EXTENTS VOLUME DEVICE-TYPE
% 1 WORK01 D3435
% NUM-OF-EXT = 1
%:20SG: PRDISC: 1 FILE RES= 3 FRE= 3 REL= 0 PAGES

```

The output of the SHOW-FILE-ATTRIBUTES command shows the catalog entry that was generated with CREATE-FILE for the file *MAX.FILE.1*, for which storage space was reserved on the private disk *WORK01*.

Example 4: Creating catalog entries for files on private disk

```

/CREATE-FILE FILE-NAME=MAX.NET-FILE.1,SUPPORT=*PUBLIC(VOL=WK0025,
                DEV-TYPE=NETSTOR)
/show-file-attr file-name=MAX.NET-FILE.1,INF=*ALL-ATTR ----- (1)
%0000000004n:CK68:$USER1.MAX.NET-FILE.1
% ----- HISTORY -----
% CRE-DATE   = NONE      ACC-DATE   = NONE      CHANG-DATE = NONE
% CRE-TIME   = NONE      ACC-TIME   = NONE      CHANG-TIME = NONE
% ACC-COUNT  = 0         S-ALLO-NUM = 0
% ----- SECURITY -----
% READ-PASS  = NONE      WRITE-PASS = NONE      EXEC-PASS  = NONE
% USER-ACC   = OWNER-ONLY ACCESS      = WRITE      ACL         = NO
% AUDIT      = NONE      FREE-DEL-D = *NONE     EXPIR-DATE = NONE
% DESTROY    = NO        FREE-DEL-T = *NONE     EXPIR-TIME = NONE
% SP-REL-LOCK= NO        ENCRYPTION = *NONE
% ----- BACKUP -----
% BACK-CLASS = A         SAVED-PAG  = COMPL-FILE VERSION    = 0
% MIGRATE    = ALLOWED
% ----- ORGANIZATION -----
% FILE-STRUC = NONE      BUF-LEN    = NONE      BLK-CONTR  = NONE
% IO(USAGE)   = READ-WRITE IO(PERF)     = STD        DISK-WRITE = IMMEDIATE
% REC-FORM    = NONE      REC-SIZE   = 0
% AVAIL       = *STD
% ----- ALLOCATION -----
% SUPPORT     = PUB      S-ALLOC    = 4         HIGH-US-PA = 0
% EXTENTS     VOLUME     DEVICE-TYPE  EXTENTS     VOLUME     DEVICE-TYPE
% 1           WK0025     NETSTOR
% NUM-OF-EXT  = 1
%:CK68: NET   : 1 FILE RES= 4 FRE= 4 REL= 4 PAGES
/CREATE-FILE FILE-NAME=MAX.NET-FILE.2,SUPPORT=*PUBLIC(STORAGE-TYPE=*NET-STORAGE) - (2)
/show-file-attr file-name=MAX.,
                SELECT=*BY-ATTR(STORAGE-TYPE=*NET-STORAGE),INF=(ALLOC=*YES) ----- (3)
%0000000004n:CK68:$USER1.MAX.NET-FILE.1
% ----- ALLOCATION -----
% SUPPORT     = PUB      S-ALLOC    = 4         HIGH-US-PA = 0
% EXTENTS     VOLUME     DEVICE-TYPE  EXTENTS     VOLUME     DEVICE-TYPE
% 1           WK0025     NETSTOR
% NUM-OF-EXT  = 1
%0000000004n:CK68:$USER1.MAX.NET-FILE.2
% ----- ALLOCATION -----
% SUPPORT     = PUB      S-ALLOC    = 4         HIGH-US-PA = 0
% EXTENTS     VOLUME     DEVICE-TYPE  EXTENTS     VOLUME     DEVICE-TYPE
% 1           CK68@0     NETSTOR
% NUM-OF-EXT  = 1
%:CK68: NET   : 2 FILES RES= 8 FRE= 8 REL= 8 PAGES

```

- (1) The output of the SHOW-FILE-ATTRIBUTES command shows the catalog entry that was generated with CREATE-FILE for the *MAX.NET-FILE.1* file which was created beforehand on the Net-Storage volume *WK0025*. The Net-Storage volume is allocated to the default pubset *CK68*. As the volume does not have the default name, the VSN and the volume type *NETSTOR* must be specified explicitly in the CREATE-FILE command.
- (2) Another *MAX.NET-FILE.2* file is created on the default Net-Storage volume of the pubset *CK68* (the VSN is *CK68@0*) using CREATE-FILE. In this case it is sufficient to specify the storage type (see the STORAGE-TYPE=*NET-STORAGE operand).
- (3) The output of the SHOW-FILE-ATTRIBUTES command shows the allocation attributes for the two files.

Example 5: Creating a catalog entry / file protection with BASIC-ACL and GUARDS

```

/cre-file file-name=max.file.2,prot=(basic-acl=*std)
/show-file-attr max.file.2,inf=(security=*yes) ----- (1)
%0000000003 :20SG:$USER1.MAX.FILE.2
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% OWNER = R W X GROUP = - - - OTHERS = - - -
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = NONE
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = NONE
% SP-REL-LOCK= NO ENCRYPTION = *NONE
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 3 REL=
/cre-file file-name=max.file.3,prot=(guards=(read=prot-a01,write=prot-a01))
/show-file-attr file-name=max.file.3,inf=(security=*yes) ----- (2)
%0000000003 :20SG:$USER1.MAX.FILE.3
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = NONE
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = NONE
% SP-REL-LOCK= NO ENCRYPTION = *NONE
% GUARD-READ = $USER1.PROT-A01
% GUARD-WRIT = $USER1.PROT-A01
% GUARD-EXEC = NONE
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 3 REL= 3 PAGES
/cre-file file-name=max.file.4,prot=(prot-attr=*from-file(max.file.3))
/show-file-attr file-name=max.file.4,inf=(security=*yes) ----- (3)
%0000000003 :20SG:$USER1.MAX.FILE.4
% ----- SECURITY -----
% READ-PASS = NONE WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = NONE
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = NONE
% SP-REL-LOCK= NO ENCRYPTION = *NONE
% GUARD-READ = $USER1.PROT-A01
% GUARD-WRIT = $USER1.PROT-A01
% GUARD-EXEC = NONE
%:20SG: PUBLIC: 1 FILE RES= 3 FRE= 3 REL= 3 PAGES

```

- (1) The output of the SHOW-FILE-ATTRIBUTES command shows the protection attributes of the file *MAX.FILE.2*. The CREATE-FILE command was used earlier to create the catalog entry and to define access control using a BASIC-ACL. All access rights (read, write, execute) were set for the file owner *USER1*. Other user IDs have no access rights.
- (2) The output of the SHOW-FILE-ATTRIBUTES command shows the protection attributes of the file *MAX.FILE.3*. The CREATE-FILE command was used earlier to create the catalog entry and to define access control using GUARDS. Read and write access from the file owner *USER1* and all other users are not permitted unless the conditions which are specified in the guard *PROT-A01* of the user *USER1* are satisfied. The file cannot be executed, since no guard was specified in the EXEC operand.
- (3) The output of the SHOW-FILE-ATTRIBUTES command shows the protection attributes of the file *MAX.FILE.4*. When the catalog entry was generated, the protection attributes were copied from *MAX.FILE.3*.

CREATE-FILE-GENERATION

Create new file generation for file generation group

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE-GENERATION-GROUP
Privileges:	STD-PROCESSING TSOS
Routing code:	\$ (with NBCONOPI=N) or E (with NBCONOPI=Y)

Function

The CREATE-FILE-GENERATION command creates a new file generation for a file generation group (which already exists). File attributes such as the generation name, the specification of the volume, the storage allocation, and performance attributes can be defined (see “Overview of functions”). The file protection and data backup attributes are defined once only for the entire file generation group using the CREATE-FILE-GROUP command. User and systems support information (see the USER-INFORMATION and ADM-INFORMATION operands) is not copied from the group entry but can be defined separately for each file generation.

A file generation group write-protected by BASIC-ACL or GUARDS cannot be extended.

For basic information on file generations and file generation groups, see the “Introductory Guide to DMS” [13].

Privileged functions

The following functions are available to nonprivileged users only if physical allocation of public space is allowed (see “Privileged functions” in the function description of the CREATE-FILE command, [page 2-380](#)):

- explicit specification (values other than the default, *STD) of a volume or volume set in the VOLUME and VOLUME-SET operands in the structure SUPPORT=*PUBLIC-DISK (STORAGE-CLASS=*NONE(...))
- absolute space reservation using SPACE=*ABSOLUTE(...) in the structure SUPPORT=*PUBLIC-DISK(...) (note that no authorization is required for SUPPORT=*PRIVATE-DISK)

Systems support personnel can supplement the user information in the file catalog with one to eight bytes of information about the file generation (ADM-INFORMATION operand).

By default, systems support (TSOS privilege) is a co-owner of all the files (and can therefore create file generations under all user logons). When SECOS is used, this co-ownership can be restricted.

In conjunction with the SECOS software product a user can allow other user IDs to act as co-owners of the file generation groups of his user ID. Co-owners of a user ID are then also allowed to create file generations under that ID.

File generations in SM pubsets

See the function description of the CREATE-FILE command, sections [“Files in SM pubsets” on page 2-381](#) and [“Physical allocation in SM pubsets” on page 2-382](#).

Work files

When a file generation is created in an SM pubset, the CREATE-FILE-GROUP command is used to define whether the file generation group is to have the work file attribute (WORK-FILE-GROUP=*YES). For more information on work files see the “Introductory Guide to DMS” [13]. If the file generation is assigned a storage class (explicitly or implicitly, see STORAGE-CLASS operand), the value specified for the WORK-FILE attribute must match the group entry of the file generation group.

File generations on tape (magnetic tape/tape cartridge)

In conjunction with a group entry on a public volume, file generations can also be stored on tape (operand SUPPORT=*TAPE). When creating or accessing the catalog entry of file generations on tape, some special points relating to the use of tapes should be taken into account.

For file generations with standard labels, details of shareability (USER-ACCESS), access type (ACCESS) and passwords are transferred from the catalog entry to the tape labels when the file is created (OPEN). The BASIC-ACL, and GUARDS protection attributes are not transferred.

No access checking is performed when a file generation is created on tape. That means that write protection even for a tape file generation that has yet to be created can be set to ACCESS=*READ. The file generation can be opened and created as an output file; write protection does not come into effect until after it has been created.

The protection attributes of the file generation group can be modified with the MODIFY-FILE-GROUP-ATTRIBUTES command. For any file generations already created on tape, the change is made only in the file catalog, not in the tape labels.

Access (OPEN) to an existing tape file generation is checked against the protection attributes stored in the file catalog. The entries in the labels are disregarded.

When a tape file generation is imported, access checking is based on the protection attributes stored in the file labels. If the tape file generation was password-protected when it was created, the same password must be specified again when it is imported, even if the file generation group is now protected by a different password or none at all.

Overview of functions

Function / Meaning	Level 1 operands	Level 2/3 operands
Specify name of file generation	GENERATION-NAME	
Specify the volume	SUPPORT	
Volume: pubset <ul style="list-style-type: none"> – Specify storage class <ul style="list-style-type: none"> – Performance requirements <ul style="list-style-type: none"> – performance attribute – type of I/O operations – Define time when data consistency is required after write operations – Define availability requirements – preferred file format – Specify volume set – Specify the volume – Specify device type – user information – systems support information 	= *PUBLIC-DISK	STORAGE-CLASS IO-ATTRIBUTES PERFORMANCE USAGE DISK-WRITE AVAILABILITY FILE-PREFORMAT VOLUME-SET VOLUME DEVICE-TYPE USER-INFORMATION ADM-INFORMATION
Volume: private disk <ul style="list-style-type: none"> – Specify device type – Request volume(s) – Storage space allocation – Separate storage of data/index for ISAM files 	= *PRIVATE-DISK	DEVICE-TYPE VOLUME SPACE DATA-SUPPORT

Table 39: Overview of CREATE-FILE-GENERATION command functions

Function / Meaning	Level 1 operands	Level 2/3 operands
Volume: tape – Specify device type – Request volume(s) – Request tape device	= *TAPE	DEVICE-TYPE VOLUME PREMOUNT-LIST
Error handling if the file already exists Normal error handling Suppress error handling	SUPPRESS-ERRORS = *NONE = *FILE-EXISTING	

Table 39: Overview of CREATE-FILE-GENERATION command functions

Format

(Part 1 of 3)

CREATE-FILE-GENERATION	Alias: CRFGN
<p>GENERATION-NAME = <filename 1..54 without-vers></p> <p>SUPPORT = *PUBLIC-DISK (...) / *PRIVATE-DISK(...) / *TAPE(...) / *NONE</p> <p> *PUBLIC-DISK (...)</p> <p> STORAGE-CLASS = *STD / <composed-name 1..8> / *NONE(...)</p> <p> *NONE(...)</p> <p> IO-ATTRIBUTES = *STD / [*PARAMETERS](...)</p> <p> [*PARAMETERS](...)</p> <p> PERFORMANCE = *STD / *HIGH / *VERY-HIGH / *USER-MAXIMUM</p> <p> USAGE = *READ-WRITE / *WRITE / *READ</p> <p> DISK-WRITE = *STD / *IMMEDIATE / *BY-CLOSE</p> <p> AVAILABILITY = *STD / *HIGH</p> <p> FILE-PREFORMAT = *BY-PUBSET-DEFAULT / *K / *NK2 / *NK4</p> <p> VOLUME-SET = *STD / *CONTROL-VOLUME-SET / <cat-id 1..4></p> <p> VOLUME = *STD / list-poss(255): <vsn 1..6></p> <p> DEVICE-TYPE = *BY-VOLUME / <device></p>	

```

,SPACE = *STD / *RELATIVE(...) / *ABSOLUTE(...)
  *RELATIVE(...)
    |
    | PRIMARY-ALLOCATION = <integer 1..2147483647>
    | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
  *ABSOLUTE(...)
    |
    | FIRST-PAGE = <integer 1..2147483647>
    | ,SIZE = <integer 1..2147483647>
,USER-INFORMATION = *NONE / <c-string 1..8 with-low>
,ADM-INFORMATION = *NONE / <c-string 1..8 with-low>
*PRIVATE-DISK(...)
  VOLUME = [ *ANY ](...) / list-poss(255): <alphanum-name 1..6>
    [ *ANY ](...)
      |
      | NUMBER-OF-DEVICES = 1 / <integer 1..9>
,DEVICE-TYPE = *BY-VOLUME-CATALOG / <device>
,SPACE = *STD / *RELATIVE(...) / *ABSOLUTE(...)
  *RELATIVE(...)
    |
    | PRIMARY-ALLOCATION = <integer 1..2147483647>
    | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
  *ABSOLUTE(...)
    |
    | FIRST-PAGE = <integer 1..2147483647>
    | ,SIZE = <integer 1..2147483647>
,DATA-SUPPORT = *SAME / [*PARAMETERS](...)
  [*PARAMETERS](...)
    |
    | DATA-DEVICE-TYPE = <device>
    | ,DATA-VOLUME = list-poss(255): <alphanum-name 1..6>
    | ,DATA-SPACE = *RELATIVE (...) / *ABSOLUTE(...)
      *RELATIVE(...)
        |
        | PRIMARY-ALLOCATION = <integer 1..2147483647>
        | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
      *ABSOLUTE(...)
        |
        | FIRST-PAGE = <integer 1..2147483647>
        | ,SIZE = <integer 1..2147483647>

```



```

*TAPE(...)
  VOLUME = *NO / [*ANY](...) / list-poss(255): <alphanum-name 1..6>
    [*ANY](...)
      NUMBER-OF-DEVICES = 1 / <integer 1..9>
    ,DEVICE-TYPE = *BY-VOLUME-CATALOG / <device>
    ,PREMOUNT-LIST = *NONE / list-poss(255): <integer 0..255>
,SUPPRESS-ERRORS = *NONE / *FILE-EXISTING

```

Operands

GENERATION-NAME = <filename 1..54 without-vers>

The name of the new file generation.

The catalog and user IDs must match those for the file generation group. Only the user's own user ID or a user ID for which the user is co-owner may be specified. Systems support personnel (TSOS privilege) may specify any user ID. The number specified for the new file generation can be either absolute or relative (to the BASE-NUMBER value of the group entry); for details see the "Introductory Guide to DMS" [13]. Regardless of how it is specified, the number of the new file generation must in absolute terms be the group entry's LAST-GEN value incremented by 1, if the value of LAST-GEN is 9999, the value for the new generation must be 1 (see also the BASE-NUMBER and LAST-GEN output fields of the SHOW-FILE-ATTRIBUTES command).

SUPPORT = *PUBLIC-DISK(...)

Indicates that the file generation is to be set up on a public disk. The pubset (related set of public disks) on which the file generation is to be created is uniquely identified by the (explicitly specified or default) catalog ID in the file name.

STORAGE-CLASS = *STD / <composed-name 1..8> / *NONE(...)

Governs the properties of the file generation's storage location. Systems support defines storage classes to facilitate automatic management of storage space in SM pubsets. A storage class is allocated certain file attributes which are implicitly set for all files assigned to that storage class. If a storage class is specified, there is no need to explicitly specify values for the VOLUME-SET, VOLUME, DEVICE-TYPE, IO-ATTRIBUTES, DISK-WRITE, FILE-PREFORMAT and AVAILABILITY operands. The WORK-FILE attribute of the storage class must match that of the group entry. The SHOW-STORAGE-CLASS command allows users to find out which storage classes of an SM pubset are available to them and which file attributes are set.

STORAGE-CLASS = *STD

A file generation in an SM pubset is assigned the default storage class from the file generation group's group entry. The default storage class is governed by the STOR-CLASS-DEFAULT operand of the CREATE-FILE-GROUP or MODIFY-FILE-GROUP-ATTRIBUTES command. If the file generation is to be created in an SF pubset or if there is no default storage class defined, the file generation is given the same attributes as with STORAGE-CLASS=*NONE with default values.

STORAGE-CLASS = <composed-name 1..8>

A file generation in an SM pubset is assigned the specified storage class. The storage class must exist and be accessible to the user at the time when the assignment is made. This specification is ignored for a file generation in an SF pubset; the file is then given the same attributes as with STORAGE-CLASS=*NONE with default values.

STORAGE-CLASS = *NONE(...)

STORAGE-CLASS=*NONE has the same effect as *STD if all the following conditions are met:

- The file generation is created on a volume set intended for permanent storage.
- A default storage class is assigned to the user ID at the SM pubset in question.
- Physical allocation is not permitted.

Only in this case are the operands PERFORMANCE, USAGE, DISK-WRITE and AVAILABILITY ignored.

In all other cases, the properties of the file generation storage location are determined by the operands of the structure *NONE(...) and no storage class is assigned to the file generation.

The WORK-FILE attribute is copied from the group entry (see the CREATE-FILE-GROUP command, WORK-FILE-GROUP operand).

For a description of the remaining operands, see the CREATE-FILE command.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed
2	0	DMS051E	Inconsistency in private disk pool or operand error detected
2	0	DMS0546	Catalog entry for specified file has reached maximum size
2	0	DMS054A	Insufficient disk space or access to disk not possible
	1	CMD0202	Syntactical or semantic error in command
	32	DMS0584	A state that does not allow the function to continue was reported during processing
	32	DMS05C7	Unexpected internal error in DMS
	64	CMD0216	Privileges error
	64	DMS0501	Requested catalog not available
	64	DMS0512	Requested catalog not available
	64	DMS051B	Requested user ID not in pubset Guaranteed messages: DMS051B, DMS0681
	64	DMS051C	User not authorized to access pubset Guaranteed messages: DMS051C, DMS0681
	64	DMS0533	Requested file not cataloged in pubset Guaranteed message: DMS0533
	64	DMS053E	File already cataloged on private volume
	64	DMS0555	Specified file name already cataloged
	64	DMS0557	Invalid volume specification
	64	DMS057A	Invalid combination of file attributes and storage class
	64	DMS057B	Invalid operand for migrated file
	64	DMS057C	Processing not possible due to HSMS error
	64	DMS057E	File has been migrated, and HSMS is not available
	64	DMS057F	Renaming of migrated file not permitted
	64	DMS0585	Error detected during catalog processing or multihost processing
	64	DMS0586	Currently not possible to access or reserve volume
	64	DMS0587	Use of specified command restricted by system administrator
	64	DMS0588	It was not possible to allocate disk space
	64	DMS05BD	Invalid combination of file and volume set attributes
	64	DMS05CC	File name already cataloged
	64	DMS05FC	Specified user ID not in HOME pubset
	64	DMS0618	Unknown storage class
	64	DMS061A	Storage class catalog could not be read
	64	DMS0683	File already exists Guaranteed message: DMS0683
	64	DMS06C4	File generation group not yet cataloged
	64	DMS06FF	BCAM connection severed
	128	DMS0506	Function not executed due to change in master
	130	DMS0524	System address space full
	130	DMS053C	No space in pubset catalog file Guaranteed messages: DMS053C
	130	DMS0582	File is currently locked or in use and cannot be processed

(Part 1 of 2)

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	130	DMS0585	Error detected during catalog processing or multihost processing
	130	DMS0586	Currently not possible to access or reserve volume
	130	DMS0588	It was not possible to allocate disk space
	130	DMS0594	Not enough virtual memory available
	130	DMS05C8	Maximum permitted number of files reached

(Part 2 of 2)

Examples

Example 1: Creating a file generation

```

/cre-file-group group-name=max.group.1,
  gen-par=(max=3,ov-opt=del-all,vol=work01,dev-type=d3435) _____ (1)
/show-file-attr group-name=max.group.1,inf=(organisation=*yes) _____ (2)
%0000000000*:20SG:$USER1.MAX.GROUP.1 (FGG)
% ----- GENERATION-INFO -----
% MAXIMUM      = 3          BASE-NUM    = 0          OVERFL-OPT = DELETE-ALL
% FIRST-GEN    = 0          LAST-GEN    = 0
% EXTENTS      VOLUME      DEVICE-TYPE
%              WORK01      D3435
%:20SG: PRDISC: 1 FILE RES=          0 FRE=          0 REL=          0 PAGES
/cre-file-gen max.group.1(*1) _____ (3)
% DMS0681 DMS ERROR '06DA' WHEN ACCESSING FILE ':20SG:$USER1.MAX.GROUP.1(*0001)
% '. FOR FURTHER INFORMATION: /HELP-MSG DMS06DA
/help-msg dms06da,lang=d _____ (4)
% DMS06DA UNZULAESSIGE KOMBINATION VON PRIVATEN UND OEFFENTLICHEN DATENTRAEGERN
% FUER EINE DATEI-GENERATIONS-GRUPPE IM ANGEGEBENEN KOMMANDO
% ? Der Benutzer versuchte, eine Generation auf einer privaten Platte
% zu erstellen und die Dateigenerationsgruppe befindet sich auf einem
% oeffentlichen Datentraeger oder es wurde versucht eine Generation
% auf oeffentlichem Datentraeger bzw. Band zu erstellen und die
% Dateigenerationsgruppe befindet sich auf privater Platte.
% ! Wenn die Dateigenerationsgruppe auf der privaten Platte
% verbleibt, muss die Generation auf der privaten Platte erstellt werden.
% Wenn sie auf einem oeffentlichen Datentraeger verbleibt, muss
% die Generation auf einem oeffentlichen Datentraeger oder auf
% einem Band bleiben.
/cre-file-gen max.group.1(*1),support=*priv(vol=work01,dev-type=d3435) _____ (5)
/cre-file-gen max.group.1(*2),support=*priv(vol=work01,dev-type=d3435)
/cre-file-gen max.group.1(*3),support=*priv(vol=work01,dev-type=d3435)
/show-file-attr f-name=max.group.1,select=by-attr(gen=y) _____ (6)
% 0*:20SG:$USER1.MAX.GROUP.1 (FGG)
% 3*:20SG:$USER1.MAX.GROUP.1(*0001)
% 3*:20SG:$USER1.MAX.GROUP.1(*0002)
% 3*:20SG:$USER1.MAX.GROUP.1(*0003)
%:20SG: PRDISC: 4 FILES RES=          9 FRE=          9 REL=          0

```

```

/show-file-attr f-name=max.group.1,inf=(organization=*yes) ----- (7)
%00000000*:20SG:$USER1.MAX.GROUP.1 (FGG)
%----- GENERATION-INFO -----
%0000000000*:20SG:$USER1.MAX.GROUP.1 (FGG)
%----- GENERATION-INFO -----
%  MAXIMUM      = 3          BASE-NUM    = 0          OVERFL-OPT = DELETE-ALL
%  FIRST-GEN    = 1          LAST-GEN   = 3
%  EXTENTS      VOLUME     DEVICE-TYPE
%  WORK01      D3435
%:20SG: PRDISC: 1 FILE RES=          0 FRE=          0 REL=          0 PAGES
/cre-file-gen max.group.1(*4),support=*priv(vol=work01,dev-type=d3435) ----- (8)
/show-file-attr max.group.1,inf=(organization=*yes) ----- (9)
%0000000000*:20SG:$USER1.MAX.GROUP.1 (FGG)
%----- GENERATION-INFO -----
%  MAXIMUM      = 3          BASE-NUM    = 0          OVERFL-OPT = DELETE-ALL
%  FIRST-GEN    = 4          LAST-GEN   = 4
%  EXTENTS      VOLUME     DEVICE-TYPE
%  WORK01      D3435
%:20SG: PRDISC: 1 FILE RES=          0 FRE=          0 REL=          0 PAGES
/show-file-attr max.group.1,select=*by-attr(gen=*yes)
0000000000*:20SG:$USER1.MAX.GROUP.1 (FGG)
%----- GENERATION-INFO -----
%  MAXIMUM      = 3          BASE-NUM    = 0          OVERFL-OPT = DELETE-ALL
%  FIRST-GEN    = 4          LAST-GEN   = 4
%  EXTENTS      VOLUME     DEVICE-TYPE
%  WORK01      D3435
0000000003*:20SG:$USER1.MAX.GROUP.1(*0004)
%----- ORGANIZATION -----
%  FILE-STRUC   = NONE      BUF-LEN    = NONE      BLK-CONTR  = NONE
%  IO(USAGE)    = READ-WRITE IO(PERF)   = STD      DISK-WRITE = IMMEDIATE
%  REC-FORM     = NONE      REC-SIZE  = 0
%  AVAIL       = *STD
%:20SG: PRDISC: 2 FILES RES=          3 FRE=          3 REL=          0 PAGES

```

- (1) A group entry is created for the file generation group *MAX.GROUP.1* on the private disk *WORK01*. A maximum of 3 file generations (*MAXIMUM=3*) are to be permitted. If this maximum is exceeded, all existing generations are to be deleted (*OVERFLOW-OPTION=*DELETE-ALL*).
- (2) The command *SHOW-FILE-ATTRIBUTES* displays the group entry.
- (3) The command *CREATE-FILE-GENERATION* is to be used to create the first file generation.
- (4) The command will be rejected. The command *HELP-MSG-INFORMATION* is used to obtain information on the error (*DMS06DA*) that occurred. The group entry is located on private disk. Since the operand *SUPPORT* was not specified for the file generation **0001*, an attempt was made to create the first file generation on public disk.
- (5) The file generations **0001*, **0002*, and **0003* are created on the private disk *WORK01*.
- (6) The command *SHOW-FILE-ATTRIBUTES* returns information on the group entry and the associated file generations (since *GENERATION=*YES*).
- (7) The command *SHOW-FILE-ATTRIBUTES* shows the attributes of the group entry that affect file generations (since *ORGANIZATION=*YES*).

- (8) The file generation *0004 is created. Since OVERFLOW-OPTION=*DELETE-ALL is defined, and the maximum has been reached, the earlier generations should have been deleted in the process.
- (9) The following SHOW-FILE-ATTRIBUTES command indicates that only the group entry and the last created generation (*0004) exist.

Example 2: Creating file generations (read password)

```

/cre-file-group group-name=max.group.2,gen-par=(max=3),
prot=(read-pass='paul') _____ (1)

/show-file-attr max.group.2,inf=(organization=*yes,security=*yes) _____ (2)
%0000000000 :20SG:$USER1.MAX.GROUP.2 (FGG)
%----- SECURITY -----
% READ-PASS = YES WRITE-PASS = NONE EXEC-PASS = NONE
% USER-ACC = OWNER-ONLY ACCESS = WRITE ACL = NO
% AUDIT = NONE FREE-DEL-D = *NONE EXPIR-DATE = 2012-03-18
% DESTROY = NO FREE-DEL-T = *NONE EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO ENCRYPTION = *NONE
%----- GENERATION-INFO -----
% MAXIMUM = 3 BASE-NUM = 0 OVERFL-OPT = CYCL-REPL
% FIRST-GEN = 0 LAST-GEN = 0
%:20SG: PUBLIC: 1 FILE RES= 0 FRE= 0 REL= 0 PAGES

/cre-file-gen max.group.2(*1) _____ (3)
% DMS0681 DMS ERROR '05D8' WHEN ACCESSING FILE ':20SG:$USER1.MAX.GROUP.2(*0001)
'. FOR FURTHER INFORMATION: /HELP-MSG DMS05D8

/help-msg dms05d8,lang=d _____ (4)
% DMS05D8 DATEI GESCHUETZT. ZUGEHORIGES KENNWORT EINGEBEN UND KOMMANDO WIEDERH
OLEN
% ? Der Benutzer versuchte fuer eine Datei, die mit einem Kennwort
% katalogisiert war, den Dateikatalog zu modifizieren, ohne dass ein
% Kennwort in der Kennworttabelle vorhanden ist.
% Das Kennwort fuer die vorher katalogisierte Datei wurde ueber das
% Kennwortkommando dem System nicht mitgeteilt.
% ! Das Dateikennwort ueber das /ADD-PASSWORD-Kommando angeben und danach das
% Kommando wiederholen.
/add-pass 'paul' _____ (5)

/cre-file-gen max.group.2(*1) _____ (6)

/cre-file-gen max.group.2(*3) _____ (7)
% DMS0681 DMS ERROR '06C7' WHEN ACCESSING FILE ':20SG:$USER1.MAX.GROUP.1(*0001)
'. FOR FURTHER INFORMATION: /HELP-MSG DMS06C7

/help-msg dms06c7,lang=d _____ (8)
% DMS06C7 UNGUELTIGE GENERATIONSNUMMER IM KOMMANDO. NUMMERN MUESSEN AUFSTIEGEND
UND LUECKENLOS SEIN. KOMMANDO NICHT AUSGEFUEHRT
% ? Generationsdateien muessen katalogisiert werden mit aufsteigend geordneten
% Generationsnummern, wobei keine Luecken in der Folge der
% Generationsnummern
% vorkommen duerfen.
% MASSNAHME : KEINE

/cre-file-gen max.group.2(*2) _____ (9)
/cre-file-gen max.group.2(*3)
/cre-file-gen max.group.2(*4)

/show-file-attr max.group.2,select=(gen=yes) _____ (10)
% 0 :20SG:$USER1.MAX.GROUP.2 (FGG)
% 3 :20SG:$USER1.MAX.GROUP.2(*0002)
% 3 :20SG:$USER1.MAX.GROUP.2(*0003)
% 3 :20SG:$USER1.MAX.GROUP.2(*0004)
%:20SG: PUBLIC: 4 FILES RES= 9 FRE= 9 REL= 9 PAGES

/show-file-attr max.group.2,inf=(organization=*yes)
%0000000000 :20SG:$USER1.MAX.GROUP.2 (FGG)
%----- GENERATION-INFO -----
% MAXIMUM = 3 BASE-NUM = 0 OVERFL-OPT = CYCL-REPL
% FIRST-GEN = 2 LAST-GEN = 4
%:20SG: PUBLIC: 1 FILE RES= 0 FRE= 0 REL= 0 PAGES

```

- (1) A group entry is created for the file generation group *MAX.GROUP.2* on the public disk *WORK01*. A maximum of 3 file generations (*MAXIMUM=3*) are to be permitted. For each new created generation that would cause the maximum to be exceeded, the oldest existing generation is to be deleted (default value *OVERFLOW-OPTION=*CYCLE-REPLACE*). The read password *PAUL* (which is replaced by "P" in logs) is defined for file protection. The protection attributes apply to the group entry and to all generations of the file generation group.
- (2) The command *SHOW-FILE-ATTRIBUTES* displays the group entry along with its protection and file generation attributes.
- (3) The command *CREATE-FILE-GENERATION* is to be used to create the first file generation.
- (4) The command will be rejected. The command *HELP-MSG-INFORMATION* is used to obtain information on the error (*DMS05D8*) that occurred. In this case, the group entry, which is updated on creating a new generation, could not be changed due to the password protection.
- (5) The password *PAUL* is entered into the password table of the task.
- (6) Generation *0001 is created.
- (7) An attempt is made to create generation *0003.
- (8) The command will be rejected. In this case, the next generation to be created must be generation *0002, since no gaps are permitted in generation numbers.
- (9) Generations *0002, *0003, and *0004 are created.
- (10) The following *SHOW-FILE-ATTRIBUTES* commands show the group entry and all existing generations: generation *0004 exceeded the defined maximum of three generations. That is why the oldest generation was deleted when creating it (default value *OVERFLOW-OPTION=*CYCLE-REPLACE*).

CREATE-FILE-GROUP

Define name and attributes of new file generation group

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE-GENERATION-GROUP
Privileges:	STD-PROCESSING TSOS
Routing code:	\$ (with NBCONOPI=N) or. E (with NBCONOPI=Y)

Function

The CREATE-FILE-GROUP command creates the catalog entry for a new file generation group on disk.

File generation groups (FGGs) consist of a group entry and the associated file generations. The group entry is created with the CREATE-FILE-GROUP command and can be modified with the MODIFY-FILE-GROUP-ATTRIBUTES command. File generations are created with the CREATE-FILE-GENERATION command and can be modified with the MODIFY-FILE-GENERATION-SUPPORT command. The file protection and data backup attributes defined in the group entry also apply to the associated file generations. User and systems support information (see USER- and ADM-INFORMATION operands) in the group entry is not “bequeathed” to the generations. This information can be defined separately for each file generation whenever a generation is created or modified.

FGGs can be created on public disks, private disks or in a nonhomogeneous mix, with the group entry on public disk and the associated file generations both on tape and on public disk. FGGs cannot be created on Net-Storage.

FGGs which are held on private volumes and for which there is no catalog entry are designated as exported or “FOREIGN” file generation groups. If such a FGG is to be re-cataloged, the group entry must be imported first, followed by the associated file generations (using the IMPORT-FILE command).

If no group entry is available for existing file generations (on private volumes or as a result of a system error in the file catalog), the CREATE-FILE-GROUP command can be used to create a new group entry for the existing file generations. This entails specifying where the generations start and finish using the FIRST-GENERATION and LAST-GENERATION operands. If the generations are on private volumes, they must then be imported with the IMPORT-FILE command.

The following must be noted for file generations on tape:

If the FGG is protected by BASIC-ACL or, in conjunction with the SECOS software product, by GUARDS (see the “SECOS” manual [35], these protection mechanisms are recorded only in the file catalog, not in the tape labels. In this case the conventional protection mechanisms - standard access control (ACCESS and USER-ACCESS) and passwords - must be used to protect tape files against unauthorized access.

Unlike an FGG which merely has ACCESS=*READ write protection, an FGG write-protected by BASIC-ACL or GUARDS cannot be extended by creating a new file generation.

Privileged functions

Systems support personnel can supplement the user information in the file catalog with one to eight bytes of information about the file generation group (ADM-INFORMATION operand).

By default, systems support (TSOS privilege) is a co-owner of all the files (and can therefore create file generation groups under all user logons). When SECOS is used, this co-ownership can be restricted. In conjunction with the SECOS software product a user can allow other user IDs to act as co-owners of the file generation groups of his user ID. Co-owners of a user ID are then also allowed to create file generations under that ID.

File generation groups in SM pubsets

A file generation group for standard files or for work files can be created in an SM pubset. This is done with the STOR-CLASS-DEFAULT operand, either explicitly (using the WORK-FILE-GROUP operand in the STOR-CLASS-DEFAULT=*NONE(...) structure or implicitly by assigning a default storage class with the corresponding WORK-FILE attribute. All the file generations in the FGG must have the same WORK-FILE attribute. The WORK-FILE attribute cannot be modified later.

Assigning a storage class simplifies the automatic management of storage space (for details see the “Introduction to System Administration” [14] or the “System-Managed Storage” manual [45]).

Overview of functions

Function / Meaning	Level 1 operands	Level 2/3 operands
Define name of the file generation group	GROUP-NAME	
Define general attributes of file generation group <ul style="list-style-type: none"> – Maximum permitted number of generations – Action when this number exceeded – Base number <ul style="list-style-type: none"> – absolute – relative to last file generation – Number of first generation – Specify device type – Specify the volume 	GENERATION-PARAMETER	MAXIMUM OVERFLOW-OPTION BASE-NUMBER =*RELATIVE-TO-LAST-GEN =*RELATIVE-TO-LAST-GEN FIRST-GENERATION DEVICE-TYPE VOLUME
Define protection attributes <ul style="list-style-type: none"> – Import protection attributes from file – File access rights – Shareability – Basic access control list – Password protection – Define passwords – Physical deletion (overwrite with binary zeros) – Monitoring of DMS accesses via SAT/System Exit Routines – Lock to prevent release of reserved storage space (disk files only) – Date of release for deletion 	PROTECTION = *PARAMETERS	PROTECTION-ATTR ACCESS USER-ACCESS BASIC-ACL <ul style="list-style-type: none"> – PASSWORD WRITE-PASSWORD/ READ-PASSWORD DESTROY-BY-DELETE AUDIT SPACE-RELEASE-LOCK FREE-FOR-DELETION

Table 40: Overview of CREATE-FILE-GROUP command functions (Part 1 of 2)

CREATE-FILE-GROUP

Function / Meaning	Level 1 operands	Level 2/3 operands
Specify default storage class No default storage class – Work file attribute	STOR-CLASS-DEFAULT STOR-CLASS-DEFAULT = *NONE	WORK-FILE-GROUP
<i>For disk files only:</i> Type and frequency of automatic data backup by ARCHIVE or HSMS – Backup frequency – Scope of backup	SAVE	BACKUP-CLASS SAVED-PAGES
HSMS storage management class Code table (XHCS)	MANAGEMENT-CLASS CODED-CHARACTER- SET	
HSMS storage management class	MANAGEMENT-CLASS	
user information	USER-INFORMATION	
systems support information	ADM-INFORMATION	

Table 40: Overview of CREATE-FILE-GROUP command functions (Part 2 of 2)

Format

(Part 1 of 2)

CREATE-FILE-GROUP

Alias: CRFGP

```

GROUP-NAME = <filename 1..47 without-gen-vers>
,GENERATION-PARAMETER = [ *GENERATION-PARAMETER ] (...)
  [ *GENERATION-PARAMETER ](...)
    MAXIMUM = <integer 1..255>
    ,OVERFLOW-OPTION = *CYCLIC-REPLACE / *REUSE-VOLUME / *DELETE-ALL /
      *KEEP-GENERATION
    ,BASE-NUMBER = *EQUAL-FIRST-GEN / <integer 1..9999>
    ,FIRST-GENERATION = 0 / <integer 0..9999>
    ,LAST-GENERATION = 0 / <integer 0..9999>
    ,DEVICE-TYPE = *BY-VOLUME / <device>
    ,VOLUME = *PUBLIC / <alphanum-name 1..6>
,PROTECTION = *STD / [*PARAMETERS](...)
  [*PARAMETERS](...)
    PROTECTION-ATTR = *BY-DEF-PROT-OR-STD / *STD / *FROM-FILE(...)
      *FROM-FILE(...)
        | FILE-NAME = <filename 1..54 without-gen>
    ,ACCESS = *BY-PROTECTION-ATTR / *WRITE / *READ
    ,USER-ACCESS = *BY-PROTECTION-ATTR / *OWNER-ONLY / *ALL-USERS
    ,BASIC-ACL = *BY-PROTECTION-ATTR / *NONE / *STD / *PARAMETERS(...) / *GROUP-R /
      *GROUP-WR / *ALL-R / *ALL-WR
    *PARAMETERS(...)
      OWNER = *NO-ACCESS / *PARAMETERS(...)
        *PARAMETERS(...)
          | READ = *NO / *YES
          | ,WRITE = *NO / *YES
      ,GROUP = *NO-ACCESS / *PARAMETERS(...)
        *PARAMETERS(...)
          | READ = *NO / *YES
          | ,WRITE = *NO / *YES
      ,OTHERS = *NO-ACCESS / *PARAMETERS(...)
        *PARAMETERS(...)
          | READ = *NO / *YES
          | ,WRITE = *NO / *YES

```

(Part 2 of 2)

```

, GUARDS = *BY-PROTECTION-ATTR / *NONE / *PARAMETERS(...)
  *PARAMETERS(...)
    |
    | READ = *NONE / <filename 1..18 without-cat-gen-vers>
    | ,WRITE = *NONE / <filename 1..18 without-cat-gen-vers>
, WRITE-PASSWORD = *BY-PROT-ATTR-OR-NONE / *NONE / <c-string 1..4> / <x-string 1..8> /
  <integer -2147483648..2147483647> / *SECRET
, READ-PASSWORD = *BY-PROT-ATTR-OR-NONE / *NONE / <c-string 1..4> / <x-string 1..8> /
  <integer -2147483648..2147483647> / *SECRET
, DESTROY-BY-DELETE = *BY-PROTECTION-ATTR / *NO / *YES
, AUDIT = *STD / *NONE / *SUCCESS / *FAILURE / *ALL
, SPACE-RELEASE-LOCK = *BY-PROTECTION-ATTR / *NO / *YES
, FREE-FOR-DELETION = *NONE / <date> / <integer 0..991231>
, SAVE = *STD / *NO / [*PARAMETERS](...)
  [*PARAMETERS](...)
    |
    | BACKUP-CLASS = *STD / *A / *B / *C / *D / *E
    | , SAVED-PAGES = *COMPLETE-FILE / *MODIFIED-PAGES
, MANAGEMENT-CLASS = *NONE / <composed-name 1..8>
, CODED-CHARACTER-SET = *USER-DEFAULT / *NONE / <name 1..8>
, USER-INFORMATION = *NONE / <c-string 1..8 with-low>
, ADM-INFORMATION = *NONE / <c-string 1..8 with-low>
, STOR-CLASS-DEFAULT = *STD / <composed-name 1..8> / *NONE(...)
  *NONE(...)
    |
    | WORK-FILE-GROUP = *NO / *YES

```

Operands

GROUP-NAME = <filename 1..47 without-gen-vers>

The name to be given to the file generation group.

Only the user's own user ID or a user ID for which the user is co-owner may be specified. Systems support personnel (TSOS privilege) may specify any user ID.

GENERATION-PARAMETER = *GENERATION-PARAMETER(...)

Specifies the attributes of the file generation group.

MAXIMUM = <integer 1..255>

Defines how many generations of the file generation group may be cataloged at the same time (see the OVERFLOW-OPTION operand).

OVERFLOW-OPTION = *CYCLIC-REPLACE / *REUSE-VOLUME / *DELETE-ALL / *KEEP-GENERATION

Specifies what should be done if the maximum permitted number of file generations (MAXIMUM operand) is exceeded. When the maximum permitted number is reached, the surplus file generations will be deleted regardless of password protection, retention period (EXPIRATION-DATE) and the permitted mode of access (ACCESS).

OVERFLOW-OPTION = *CYCLIC-REPLACE

The current oldest generation is deleted and its storage space, or the tapes it occupies, are released. In the catalog, the entries in the *LAST-GEN* and *FIRST-GEN* output fields (most recent/oldest existing generations) are updated.

OVERFLOW-OPTION = *REUSE-VOLUME

The effect of OVERFLOW-OPTION=*REUSE-VOLUME depends on the storage medium:

For FGGs on public disks: the oldest generation will be deleted and its storage space returned to the system; the group entry is updated (see OVERFLOW-OPTION=*CYCLIC-REPLACE).

For FGGs on private disks: the new generation is created and the oldest generation deleted, the volume being used for storage of the new generation. If the generation which is being deleted extends over several disks, the new generation is cataloged only on the first of these disks. The group entry is updated accordingly. Since the old generation is not deleted until the new generation has been created, lack of storage space on the disk may mean that it is impossible to create the new generation, even though OVERFLOW-OPTION=*REUSE-VOLUME is specified.

For FGGs on tape: the oldest generation is deleted from the catalog, and the new generation is created on the tape thus released. The group entry is updated accordingly. OVERFLOW-OPTION=*REUSE-VOLUME is not permitted for file generation groups on multifile tapes (file sets).

OVERFLOW-OPTION = *DELETE-ALL

All generations of the FGG are deleted, and the new generation becomes the oldest one in the new series. The group entry is updated accordingly.

OVERFLOW-OPTION = *KEEP-GENERATION

The file generations are not automatically deleted. The oldest file generations which overshoot the maximum will not be deleted until OVERFLOW-OPTION or BASE-NUMBER is modified (MODIFY-FILE-GROUP-ATTRIBUTES command).

BASE-NUMBER = *EQUAL-FIRST-GEN / <integer 0..9999>

Defines a reference point/a base generation to which all relative generation numbers refer. The names of the file generations can be specified with absolute (*n) or relative (±n) generation numbers.

BASE-NUMBER = *EQUAL-FIRST-GEN

The starting base is the value specified by FIRST-GENERATION.

FIRST-GENERATION = 0 / <integer 1..9999>

Specifies the absolute generation number of the oldest existing file generation. The operand should be used only in combination with the LAST-GENERATION operand in order to reconstruct the group entry of a file generation group. This needs to be done in order to work with existing file generations for which no group entry exists (see the function description on [page 2-442](#)).

VOLUME = *PUBLIC / <alphanum-name 1..6>

VSN of the public/private disk on which the file generation is to be created.

VOLUME = *PUBLIC

The file generation group will be created on a public disk. The generations which belong to it can also be set up on tapes or tape cartridges.

VOLUME = <alphanum-name 1..6>

The file generation group will be created on a private disk. The generations which belong to it must also be set up on a private disk.

DEVICE-TYPE = *BY-VOLUME / <device>

The type of device to which the required disk is assigned (if the file generation group is to be created on a private disk).

DEVICE-TYPE = *BY-VOLUME

File generation groups on public volumes:

The system itself determines the device type. If the file generation group is being set up on a public disk, (VOLUME=*PUBLIC), then the only permissible specification is *BY-VOLUME.

DEVICE-TYPE = <device>

File generation groups on private volumes:

If the file generation group is being set up on a private disk, then the device type must be explicitly specified here (*BY-VOLUME is not allowed). Only device types known in the system are accepted. In interactive mode, DEVICE-TYPE=? calls up a list of the available device types. If at least one volume ID is specified with VOLUME, each specification of a disk device type which is known to the system is handled like the STDDISK specification. The permissible specifications for DEVICE-TYPE can also be found in the device table in [section "Device type table" on page 1-82](#) (device type column)).

LAST-GENERATION = 0 / <integer 0..9999>

Specifies the absolute generation number of the most recent existing file generation. The operand should be used only in combination with the FIRST-GENERATION operand in order to reconstruct the group entry of a file generation group. This needs to be done in order to work with existing file generations for which no group entry exists (see the function description on [page 2-442](#)).

PROTECTION = *STD / *PARAMETERS(...)

The protection attributes for the file.

PROTECTION = *STD

The values supplied by default protection are employed for the protection attributes. The AUDIT and FREE-FOR-DELETION protection attributes are always set to *NONE. If default protection is not active, the system default values for the operands of the *PARAMETERS structure are set.

PROTECTION-ATTR=	*FROM-FILE	*STD	*BY-DEF-PROT-OR-STD
		Default prot. not active	
Protection attribute		(System standard values)	
ACCESS	value from reference file	WRITE	
USER-ACCESS		OWNER-ONLY	
BASIC-ACL		NONE	
DESTROY-BY-DELETE		NO	
PASSWORD		NONE	
SPACE-RELEASE-LOCK		NO	
READ-PASSWORD		values from default protection	
WRITE-PASSWORD			
FREE-FOR-DELETION			
AUDIT			

Table 41: Effects of PROTECTION-ATTR on the protection attributes in the case of CREATE-FILE-GROUP

PROTECTION = *PARAMETERS(...)

The file generation group is given the specified protection attributes. For descriptions of the PROTECTION-ATTR, ACCESS, USER-ACCESS, BASIC-ACL, GUARDS, WRITE-PASSWORD, READ-PASSWORD, DESTROY-BY-DELETE, AUDIT, SPACE-RELEASE-LOCK and FREE-FOR-DELETION operands, see the corresponding operand descriptions for the CREATE-FILE command.

Note the following

Since file generations are not executable (program or procedure), they cannot be protected with an execute password (EXEC-PASSWORD). Likewise for the same reason no execute permission can be set within BASIC-ACL or GUARDS.

SAVE = ..., MANAGEMENT-CLASS = ..., CODED-CHARACTER-SET = ..., USER-INFORMATION = ..., ADM-INFORMATION = ...

See the description of the corresponding operand in the CREATE-FILE command.

STOR-CLASS-DEFAULT = *STD / <composed-name 1..8> / *NONE(...)

The operand is evaluated only for file generation groups in SM pubsets. Governs the properties of the storage location if no explicit specification is made when the file generation is created. Only the “work file” attribute is copied from a specified storage class into the group entry, and it then governs whether the associated generations are work files or standard files.

The only specification allowed for a file generation group in an SF pubset is STOR-CLASS-DEFAULT=*NONE.

The SHOW-STORAGE-CLASS command allows users to find out which storage classes of an SM pubset are available to them and which file attributes are set.

Assigning a storage class simplifies the automatic management of storage space in an SM pubset (for details see the “Introduction to System Administration” [14] or the “System-Managed Storage” manual [45]).

STOR-CLASS-DEFAULT = *STD

The pubset-specific default storage class from the user entry is used as the default storage class for the file generation group. If the file generation group is to be created in an SF pubset or if there is no default storage class defined in the user entry, the file generation group is given the same attributes as with STOR-CLASS-DEFAULT=*NONE with default values.

STOR-CLASS-DEFAULT = <composed-name 1..8>

A file generation group in an SM pubset is assigned the specified storage class. The storage class must exist and be accessible to the user at the time when the assignment is made. This specification is ignored for a file generation group in an SF pubset; the FGG is then given the same attributes as with STOR-CLASS-DEFAULT=*NONE with default values.

STOR-CLASS-DEFAULT = *NONE(...)

This value has the same effect as *STD if all the following conditions are met:

- The file generation group is created on a volume set intended for permanent storage.
- A default storage class is assigned to the user ID at the SM pubset in question.
- Physical allocation is not permitted.

Only in this case is the WORK-FILE-GROUP operand ignored.

In all other cases, no storage class is assigned to the file generation group.

WORK-FILE-GROUP = *NO / *YES

The operand is evaluated only for file generation groups in SM pubsets. Specifies whether the associated file generations are work files which systems support is allowed to delete at a time which it defines.

WORK-FILE-GROUP = *NO

The associated file generations are not to be work files.

WORK-FILE-GROUP = *YES

The associated file generations are to be work files.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
2	0	CMD0001	Command executed
	0	DMS06CA	Command executed, but invalid specification in BASE-NUMBER operand ignored
	1	CMD0202	Syntactical or semantic error in command
	32	DMS0584	A state that does not allow the function to continue was reported during processing
	32	DMS05C7	Unexpected internal error in DMS
	64	CMD0216	Privileges error
	64	DMS0501	Requested catalog not available
	64	DMS0512	Requested catalog not available
	64	DMS051B	Requested user ID not in pubset
	64	DMS051C	User not authorized to access pubset
	64	DMS053E	File already cataloged on private volume
	64	DMS057B	Invalid operand for migrated file

(Part 1 of 2)

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	64	DMS057C	Processing not possible due to HSMS error
	64	DMS057E	File has been migrated, and HSMS is not available
	64	DMS057F	Renaming of migrated file not permitted
	64	DMS0585	Error detected during catalog processing or multihost processing
	64	DMS0586	Currently not possible to access or reserve volume
	64	DMS0587	Use of specified command restricted by system administrator
	64	DMS0588	It was not possible to allocate disk space
	64	DMS05B5	Guard not available
	64	DMS05CC	File name already cataloged
	64	DMS05FC	Specified user ID not in HOME pubset
	64	DMS060D	Invalid file name for reference file (PROTECTION-ATTR)
	64	DMS0613	Unknown management class
	64	DMS0618	Unknown storage class
	64	DMS061A	Storage class catalog could not be read
	64	DMS06FF	BCAM connection severed
	128	DMS0506	Function not executed due to change in master
	130	DMS0524	System address space full
	130	DMS053C	No space in pubset catalog file Guaranteed messages: DMS053C
	130	DMS0585	Error detected during catalog processing or multihost processing
	130	DMS0586	Currently not possible to access or reserve volume
	130	DMS0594	Not enough virtual memory available
	130	DMS05C8	Maximum permitted number of files reached

(Part 2 of 2)

Examples

Examples of the use of the CREATE-FILE-GROUP command are given in the examples for the CREATE-FILE-GENERATION command.

CREATE-GS-COMPLEX

Create GS complex

Description status:	GSMAN V19.0A
Functional area:	Global storage administration
Domain:	DEVICE
Privileges:	TSOS

Function

Systems support uses the CREATE-GS-COMPLEX command to create a new GS complex consisting of the local GS server and a GS unit. Following command processing, the specified GS unit at the local GS server has the status `ONLINE/DETACHED`.

The command is rejected if the GS server and/or the specified GS unit already belongs to an existing GS complex.

The MODIFY-GS-COMPLEX command can be used to add or remove further GS servers and/or GS units. The SHOW-GS-COMPLEX-CONFIGURATION command supplies information on the configuration of the GS complexes.

For more details on administering global storage, see the “Introduction to System Administration” [14].

Format

CREATE-GS-COMPLEX

GS-UNIT = <integer 1..2>

Operand

GS-UNIT = <integer 1..2>

Number of the GS unit

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD0202	Syntax error
	32	CMD0221	Internal error
	64	CMD0216	Privileges error
	64	EGC3001	GS unit not available
	64	EGC3002	GS unit already in GS complex
	64	EGC3003	GS server already in GS complex
	128	EGC2050	GS function not available
	128	EGC2051	SVP function not available
	128	EGC2052	Internal SVP error

CREATE-GS-PARTITION

Create partition in GS

Description status:	GSMAN V19.0A
Functional area:	Caching media control Global storage administration
Domain:	DEVICE
Privileges:	TSOS

Function

The CREATE-GS-PARTITION command is used by systems support to create a partition within the global storage (GS) medium. Global storage (GS) is a high-capacity, non-volatile storage medium used for buffer storage of data (for information on the administration of global storage see also the “Introduction to System Administration” [14]).

A partition with the specified name (PARTITION-ID) is physically set up in GS on the basis of the size (SIZE operand), mode of operation (MODE operand) and location (LOCATION operand) specified in the command. The configuration data for the partition is likewise created in GS.

In global GS operation in an XCS network the command applies to all the nodes of the network. The GSMAN subsystem is not available until after XCS has been started.

Systems support can use the SHOW-GS-STATUS command to check up on the current GS configuration. A partition can be deleted again with the DELETE-GS-PARTITION command.

Possible users of the GS partition

GS partitions are currently used by the following system components and products which each define their own name spaces for their partitions (an “x” in the table corresponds to exactly one freely selectable character). For details see the listed manuals:

Component or product	Name of GS partition	Described in
DAB	DABxxxxx	“DAB” [5]
GSVOL	GSVxxxxx	“Introduction to System Administration” [14]
NSM	NSM	“HIPLEX MSCF” [25]
VM2000	VIRTGSxx	“VM2000” [50]

Format

CREATE-GS-PARTITION

```
PARTITION-ID = <name 1..8>
,SIZE = <integer 1..67108862 Mbyte>
,MODE = *DUAL / *MONO(...)
    *MONO(...)
    |   GS-UNIT = <integer 1..2>
,LOCATION = *ANY / <integer 2..134217727 Mbyte>
```

Operands

PARTITION-ID = <name 1..8>

Name of the GS partition to be created. The GS user defines what name space can be used (see [page 2-455](#)).

If a partition with this name already exists, the command will be rejected.

SIZE = <integer 1..67108862 Mbyte>

Specifies the size of the partition in megabytes. The maximum size is governed by the free storage areas of the GS unit. It can be checked with the command SHOW-GS-STATUS SELECT=*FREE-GS(...). In the case of MODE=*DUAL the location is also crucial.

2 MB per GS unit is reserved for GS Manager administration data.

MODE = *DUAL / *MONO(...)

Specifies whether the partition is to be created on only one of the GS units (mono mode) or on both (dual mode).

The default is *DUAL, i.e. the partition is set up on both GS units, with identical sizes and locations (dual mode).

MODE = *MONO(...)

The partition is created only on the GS unit specified next (mono mode).

GS-UNIT = <integer 1..2>

Number of the GS unit.

LOCATION = *ANY / <integer 2..134217727 Mbyte>

Governs the position of the partition within the GS unit, specified as an offset in megabytes from the beginning of the GS unit (with MODE=*MONO) or from the beginning of GS unit 1 (with MODE=*DUAL).

The default is *ANY, i.e.

- a mono partition begins at the first available point in the GS unit which will support the requested size.
- a dual partition begins at the last available point in the two GS units which will support the requested size.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
	1	CMD0202	Syntax error
	32	CMD0221	Internal error
	64	CMD0216	No authorization to invoke command
	64	EGC0112	No GS available
	64	EGC0201	The specified partition already exists
	64	EGC0202	Not enough space in GS unit to create partition
	64	EGC0203	Specified location in GS unit is possible
	64	EGC0204	GS unit space not sufficient for creation of the dual partition
	64	EGC0205	Specified GS unit is not available
	128	EGC0010	GSMAN subsystem is not ready
	128	EGC0110	Command temporarily not executable
	128	EGC0210	Maximum number of partitions reached

CREATE-GS-VOLUME

Create GS volume in GS partition

Description status:	GSVOL V1.3B
Functional area:	Caching media control Global storage administration
Domain:	DEVICE SYSTEM-TUNING STORAGE-MANAGEMENT
Privileges:	TSOS

Function

The CREATE-GS-VOLUME command allows systems support to create a GS volume in a GS partition set up beforehand. Systems support also defines the device mnemonic and format of the volume. When the GS volume is created, the contents of the GS partition are deleted, since a GS volume can later no longer be deleted with VOLIN (VOLIN does not format a GS volume physically).

The command applies equally to all the systems within an XCS network. This guarantees that the device mnemonic of a GS volume is the same on all the systems in the XCS network. Consequently the command needs to be issued just once for each new GS volume created.

The DELETE-GS-VOLUME command deletes GS volumes. The SHOW-GS-VOLUME-ATTRIBUTES command allows systems support to find out about available GS volumes.

Format

CREATE-GS-VOLUME

PARTITION-ID = <name 3..8>

,DEVICE-UNIT = <x-text 4..4>

,FORMAT = *NK4 / *K / *NK2

Operands**PARTITION-ID = <name 3..8>**

Specifies the name of the partition on which the GS volume is to be set up.
The name must begin with the string *GSV*.

DEVICE-UNIT = <x-text 4..4>

Defines the device mnemonic of the GS volume being created (the device mnemonic must fall within the range 0300 - 03FF).

FORMAT = *NK4 / *K / *NK2

Specifies the format of the GS volume.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	1	CMD0202	Syntax error in command
	32	CMD0221	Internal error
	64	CMD0216	No authorization to issue command
	64	NDG0002	Device mnemonic already allocated
	64	NDG0005	Device mnemonic not in allowable range 0300 - 03FF
	64	NDG0006	Error accessing partition
	64	NDG0008	Partition already occupied by another volume
	64	NDG0009	Partition not yet created
	64	NDG0010	Name of partition incorrect

Example

```
/CREATE-GS-VOLUME PARTITION-ID=GSV01,DEVICE-UNIT=0314,FORMAT=*NK2
```

CREATE-ISAM-POOL

Create ISAM pool or attach task to ISAM pool

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING

Function

The CREATE-ISAM-POOL command enables the user to create an ISAM pool or to link his or her job (task) with any existing ISAM pool that was created with system-wide scope.

Every ISAM pool is uniquely identified by the following attributes, which are specified at the time that the ISAM pool is created:

- The name of the ISAM pool (POOL-NAME operand)
- The catalog ID of the subset (CAT-ID operand)
- Its scope of applicability (SCOPE operand)
- The type of buffering (WRITE-IMMEDIATE operand)
- Its size (SIZE operand).
- Its performance attribute (RESIDENT operand)

If the task is to be linked with an existing ISAM pool, there should be no conflict between the specifications in the CREATE-ISAM-POOL command and the attributes of the existing ISAM pool. The existing attributes of a previously created ISAM pool can be displayed by using the SHOW-ISAM-POOL-ATTRIBUTES command.

Note for ISAM pools

As of BS2000/OSD V6.0B cross-task ISAM pools are automatically created on a file-specific basis when a file is opened in a data space. The CREATE-ISAM-POOL command with SCOPE=*USER-ID/*USER-GROUP is only supported for reasons of compatibility and has the same effect as SCOPE=*HOST-SYSTEM (for details on ISAM pools in data spaces see the “Introductory Guide to DMS” [13]).

NK-ISAM files in NK2 and NK4 format

NK-ISAM files in both NK2 format as well as NK4 format can be processed via the ISAM pool. When the first NK-ISAM file to be processed is opened, the ISAM pool is formatted in accordance with that file. If an NK-ISAM file in another format is then opened via the same ISAM pool, the ISAM pool is extended dynamically by an extent equal to its previous size and is formatted according to the new file to be processed. In other words, after the

extension, the ISAM pool will consist of one extent for processing NK2-ISAM files and one for processing NK4-ISAM files, thus doubling the actually required memory space to twice the SIZE specification.

To prevent the resulting degradation in performance, it is therefore advisable to only process files of the same format via an ISAM pool.

A detailed description of ISAM pools can be found in the “Introductory Guide to DMS” [13].

Format

CREATE-ISAM-POOL
<pre> POOL-NAME = <name 1..8> , CAT-ID = *<u>DEFAULT-PUBSET</u> / <cat-id 1..4> , SCOPE = *<u>TASK</u> (...) / *<u>HOST-SYSTEM</u>(...) / *<u>USER-ID</u>(...) / *<u>USER-GROUP</u>(...) *<u>TASK</u>(...) WRITE-IMMEDIATE = *<u>NO</u> / *YES *<u>HOST-SYSTEM</u>(...) WRITE-IMMEDIATE = *<u>YES</u> / *NO CREATION-MODE = *<u>ANY</u> / *NEW *<u>USER-ID</u>(...) WRITE-IMMEDIATE = *<u>YES</u> / *NO CREATION-MODE = *<u>ANY</u> / *NEW *<u>USER-GROUP</u>(...) WRITE-IMMEDIATE = *<u>YES</u> / NO CREATION-MODE = *<u>ANY</u> / NEW , SIZE = *<u>STD</u> / <integer 32..32767 2Kbyte> , RESIDENT = *<u>NO</u> / *YES </pre>

Operands

POOL-NAME = <name 1..8>

Assigns a name to the ISAM pool which is to be created or which already exists, and to which the task is to be linked. Together with its catalog ID and scope, this allows the pool to be uniquely identified.

CAT-ID = *DEFAULT-PUBSET / <alphanum-name 1..4>

Specifies the catalog ID of the pubset to which the ISAM pool is to be assigned.

The ISAM pool will be set up on the host computer to which this pubset belongs.

The catalog ID can – as in the file name – be regarded as part of the name, i.e. different catalog IDs identify different ISAM pools.

CAT-ID = *DEFAULT-PUBSET

The ISAM pool is assigned to the catalog that was set with the ISPLDFC system parameter (**ISAM-POOL-DEFAULT-CATID**):

X'00': default catalog ID from the user entry (see the SHOW-USER-ATTRIBUTES command, output field *DEFAULT-PUBSET*)

X'01': catalog ID of the home pubset

CAT-ID = <alphanum-name 1..4>

The catalog ID of the pubset to which the ISAM pool is to be assigned.

SCOPE = *TASK(...) / *USER-ID(...) / *USER-GROUP(...) / *HOST-SYSTEM(...)

Specifies the scope of the ISAM pool and governs the possible users:

- task-local, i.e. only for the creating task
- cross-task, for all the tasks of the system

The scope identifies the ISAM pool uniquely, i.e. ISAM pools that have the same name and same catalog ID but different scopes are not the same.

SCOPE = *TASK(...)

The ISAM pool can only be used by the calling job: it is task-local.

WRITE-IMMEDIATE = *NO / *YES

Specifies whether the blocks which are buffered in the ISAM pool are to be written out immediately after any change, or only when the buffer is required for another block.

WRITE-IMMEDIATE = *NO

For files which are to be processed in this pool, any WRITE-IMMEDIATE specification in an ADD-FILE-LINK command, or as specified in the FCB in the program, will apply. If these contain no specification, then *NO will be applicable.

WRITE-IMMEDIATE = *YES

For all the files to be processed in this pool, the WRITE-IMMEDIATE function will apply: any amended blocks will immediately be written back to the disk.

It should be noted that this will greatly increase the input/output rate, and performance will be reduced.

SCOPE = *HOST-SYSTEM(...)

The ISAM pool is cross-task and can be used by any job (task).

WRITE-IMMEDIATE = *YES / *NO

Specifies whether the blocks which are buffered in the ISAM pool are to be written out immediately after any change, or only when the buffer is required for another block.

WRITE-IMMEDIATE = *YES

Activates the WRITE-IMMEDIATE function for all files processed in this pool. In other words, updated blocks are written back to disk immediately.

WRITE-IMMEDIATE = *NO

Modified blocks are not written back to disk until the buffer is required for another block. WRITE-IMMEDIATE=*NO must be defined in the ADD-FILE-LINK command for any file to be processed via this ISAM pool; otherwise, the file cannot be opened.

CREATION-MODE = *ANY / *NEW

Specifies whether the user wishes to establish a new ISAM pool, or whether a link to an ISAM pool with the same name and catalog ID, which may already exist, may be established.

CREATION-MODE = *ANY

If there is already a cross-task ISAM pool with the same name and catalog ID, the job is linked to this pool. The specified SIZE of the pool is **not** taken into account. If no such ISAM pool currently exists, a new pool is set up with the size defined in SIZE.

CREATION-MODE = *NEW

A new ISAM pool is to be created. However, if a cross-task ISAM pool with the same name and catalog ID already exists, the command will be rejected with an error message.

SCOPE = *USER-ID(...) / *USER-GROUP(...)

The ISAM pool can be used by all jobs (tasks) which belong to the same user group as the task which created it. For a description of the lower-level operands WRITE-IMMEDIATE and CREATION-MODE see SCOPE=*USER-ID.

SIZE = *STD / <integer 32..32767 2Kbyte>

Defines the size of the new ISAM pool to be created.

SIZE = *STD

A task-local ISAM pool is set up with the default size defined in the ISAM parameter LCLPS in the parameter service. An ISAM pool which is cross-task is set up with the size defined in the ISAM parameter GLBPS.

SIZE = <integer 32..524288>

Defines the size of the ISAM pool to be created in units of 2 Kbytes (PAM pages). The following are possible size specifications:

$32 \leq \text{SIZE} \leq 32767$ for ISAM pools which are cross-task

$32 \leq \text{SIZE} \leq 8192$ for task-local ISAM pools

The specified SIZE is rounded internally if required. The rounded value for SIZE is shown in the output of the SHOW-ISAM-POOL-ATTRIBUTES command.

The minimum size of 32 PAM pages means that files with logical blocks of no more than STD(6) can be processed. An ISAM pool of 96 PAM pages is required for files with logical blocks of size STD(16).

NK-ISAM files in both NK2 format as well as NK4 format can be processed via the ISAM pool. When the first NK-ISAM file to be processed is opened, the ISAM pool is formatted in accordance with that file. If an NK-ISAM file in another format is then opened via the same ISAM pool, the ISAM pool is extended dynamically by an extent equal to its previous size and is formatted according to the new file to be processed. To prevent a degradation in performance, only files of one format should be processed via an ISAM pool.

RESIDENT = *NO / *YES

Performance attribute of the ISAM pool.

Specifies whether the ISAM pool is to be created as a memory-resident pool if the appropriate resources are currently available.

If the task is to be linked to an existing ISAM pool, this specification must match the performance attribute defined for that pool.

RESIDENT = *NO

The memory pages of the ISAM pool are pageable.

RESIDENT = *YES

This specification is only possible for users who have the DMS tuning privilege

CONCURRENT-USE or EXCLUSIVE-USE for the pubset (see the output field

DMS-TUNING-RESOURCES in the output of the SHOW-USER-ATTRIBUTES command).

Depending on the currently available resources, the memory pages of the ISAM pool are kept resident in memory as far as possible.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed without errors
	32	DMS0A17	Internal system error
	64	DMS0A0E	Syntax error in ISAM pool command
	64	DMS0A11	Specified catalog ID does not exist
	64	DMS0A13	Specified pool name is syntactically invalid
	64	DMS0A15	Specified ISAM pool already exists
	64	DMS0A18	Size of ISAM pool invalid
	64	DMS0A1E	Not authorized for resident pool
	64	DMS0A1F	RESIDENT parameters incompatible
	64	DMS0A20	Parameters not permitted for remote BS2000 version
	64	DMS0A21	ISAM pool limit reached
	64	DMS0A22	User group does not exist
	130	DMS0A12	Specified catalog ID not available
	130	DMS0A14	Shortage of memory space for CREATE-ISAM-POOL

Example

Connecting a task to several ISAM pools

```

/show-isam-pool-attr pool=*all _____ (1)
%
% CATID   POOLNAME SCOPE           WROUT   SIZE  EXTENTS  RESIDENT
%-----
% 10SB    SDFPOOLN  TASK           NO      32    --/--    NO
%
%

/cre-isam-pool pool-name=poolab01,scope=*host _____ (2)
/cre-isam-pool pool-name=poolab01,scope=*task _____ (3)

/show-isam-pool-attr pool=*all _____ (4)
%
% CATID   POOLNAME SCOPE           WROUT   SIZE  EXTENTS  RESIDENT
%-----
% 10SB    SDFPOOLN  TASK           NO      32    --/--    NO
% N      POOLAB01  HOST           YES     512   --/--    NO
% N      POOLAB01  TASK           NO      512   --/--    NO
%

/show-isam-pool-attr pool=poolab01(scope=host),inf=*user-and-attr _____ (5)
%
% CATID   POOLNAME SCOPE           WROUT   SIZE  EXTENTS  RESIDENT
%-----
% N      POOLAB01  HOST           YES     512   --/--    NO
%
%----- ANGESCHLOSSENE TASKS -----
%
%                               TSN = 1EUW
%                               TSN = 1EUE
%-----

/show-isam-pool-attr pool=poolab01(scope=*task),inf=*user-and-attr _____ (6)
%

```

```

% CATID      POOLNAME SCOPE                WROUT  SIZE  EXTENTS  RESIDENT
%-----
% N          POOLAB01 TASK                NO     512   --/--   NO
%
%----- ANGESCHLOSSENE TASKS -----
%
%                                TSN = 1EUE
%-----
%
%
```

- (1) Returns information on all ISAM pools to which the task is connected. There is one existing task-local ISAM pool: *SDFPOOLN*.
- (2) Connects the task to the cross-task ISAM pool *POOLAB01*.
- (3) Connects the task to the task-local ISAM pool *POOLAB01*. The task-local ISAM pool has the same name as the host-specific ISAM pool; however, the name is considered unique, since the scope differs.
- (4) Returns information on all ISAM pools to which the task is connected. The output shows that there are three existing ISAM pools: one named *SDFPOOLN* and two with the name *POOLAB01*, of which one is cross-task, and one is task-local. It is evident that no file has been processed with these ISAM pools, since the *EXTENTS* output column does not contain any formatting information.
- (5) Shows the attributes and the connected task for the cross-task ISAM pool *POOLAB01*. Two tasks are connected to this pool: the ISAM pool which was created earlier by the task with TSN *1EUW*, and the user's own TSN *1EUE*, which is shown as the second task (see also Point 6).
- (6) Shows the attributes and the connected task for the task-local ISAM pool *POOLAB01*. As expected, only the users' own task (TSN *1EUE*) can be connected to the pool in this case. This ISAM pool was created using the CREATE-ISAM-POOL command (see Point 3).

CREATE-JV

Define name and attributes for new job variable

Description status:	JV V15.1A
Functional area:	Job variables
Domain:	JOB-VARIABLES
Privileges:	STD-PROCESSING HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION
Routing code:	\$ (with NBCONOPI=N) or J (with NBCONOPI=Y)

This function is only available to the user if the chargeable software product JV has been loaded as a subsystem.

Function

The CREATE-JV command generates the catalog entry for a JV whose value is undefined until the first change. The user defines the name of the JV and the following protection attributes:

- read-only or read/write access (ACCESS operand); part of standard access control
- access by foreign user IDs (USER-ACCESS operand); part of standard access control
- explicit assignment of access rights (BASIC-ACL operand); extended access control
- protection by guards (GUARDS operand)
- additional protection via passwords (READ- and WRITE-PASSWORD operands)
- HSMS management class (MANAGEMENT-CLASS operand)

The protection attribute “retention period” is set implicitly. It can only be assigned via the MODIFY-JV-ATTRIBUTES command (RETENTION-PERIOD operand). By default, no retention period is defined for newly created JVs, i.e. the output field *EXPIR-DATE* in the catalog entry contains the creation date (output field *CRE-DATE*). Currently, the output field *EXPIR-TIME* is always set to the value 00:00:00. For the creation time (output field *CRE-TIME*), the actual time of creation is entered.

For temporary JVs, only the preset values are allowed, i.e. they are not shareable, can always be overwritten and cannot be protected by passwords. The protection function BASIC-ACL is likewise not supported. Since only the creating job can access temporary JVs up to LOGOFF processing, no protection is required against foreign access.

When creating permanent JVs, the user cannot exceed the maximum permitted number specified in the user entry for the relevant subset. If the number of permanent JVs is the same as the maximum permitted number, any further CREATE-JV commands for creating a permanent JV are rejected (see the *JOBVARIABLES* and *JV-NUMBER-LIMIT* output fields of the SHOW-USER-ATTRIBUTES command).

Privileged functions

By default, systems support (TSOS privilege) is a co-owner of all job variables (and can therefore create job variables under all user logons). When SECOS is used, this co-ownership can be restricted for permanent job variables.

Format

(Part 1 of 2)

CREATE-JV	Alias: CRJV
<pre> JV-NAME = <filename 1..54 without-gen-vers> ,PROTECTION = *STD / [*PARAMETERS](...) [*PARAMETERS](...) PROTECTION-ATTR = *BY-DEF-PROT-OR-STD / *STD ,ACCESS = *BY-PROTECTION-ATTR / *WRITE / *READ ,USER-ACCESS = *BY-PROTECTION-ATTR / *OWNER-ONLY / *ALL-USERS ,BASIC-ACL = *BY-PROTECTION-ATTR / *NONE / *STD / [*PARAMETERS](...) [*PARAMETERS](...) OWNER = *NO-ACCESS / [*PARAMETERS](...) [*PARAMETERS](...) READ = *NO / *YES ,WRITE = *NO / *YES ,GROUP = *NO-ACCESS / [*PARAMETERS](...) [*PARAMETERS](...) READ = *NO / *YES ,WRITE = *NO / *YES ,OTHERS = *NO-ACCESS / [*PARAMETERS](...) [*PARAMETERS](...) READ = *NO / *YES ,WRITE = *NO / *YES </pre>	

```

, GUARDS = *BY-PROTECTION-ATTR / *NONE / [*PARAMETERS](...)
    [*PARAMETERS](...)
        READ = *NONE / <filename 1..18 without-cat-gen-vers>
        , WRITE = *NONE / <filename 1..18 without-cat-gen-vers>
    , READ-PASSWORD = *BY-PROTECTION-ATTR / *NONE / *SECRET /
        <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
    , WRITE-PASSWORD = *BY-PROTECTION-ATTR / *NONE / *SECRET /
        <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
, MANAGEMENT-CLASS = *NONE / <composed-name 1..8>
, SUPPRESS-ERRORS = *NONE / *JV-EXISTING

```

Operands

JV-NAME = <filename 1..54 without-gen-vers>

Name of the job variable JV to be created.

Nonprivileged users can create job variables under their own user ID only. The name of a temporary JV begins with the character (# or @) specified in the TEMPFIL system parameter. The internal name is always output by the system (e.g. SHOW-JV-ATTRIBUTES, system messages).

PROTECTION = *STD / *PARAMETERS(...)

Protection attributes of the JV.

PROTECTION = *STD

The protection attributes listed below are assigned the values supplied by default protection.

If default protection is not active, the system default values for the operands of the *PARAMETERS structure are set.

PROTECTION-ATTR=	*STD	*BY-DEF-PROT-OR-STD	
		Default protection not active	Default protection active
Protection attribute	(System standard values)		
ACCESS		WRITE	values from default protection
USER-ACCESS		OWNER-ONLY	
BASIC-ACL		NONE	
PASSWORD		NONE	
READ-/WRITE-PASSWORD		NONE	

Table 42: Effects of PROTECTION-ATTR on protection attributes in the case of CREATE-JV

PROTECTION = *PARAMETERS(...)

The protection attributes of the JV are set as follows. If the default value *NONE or *NO is specified for an attribute, the corresponding protection function is not activated. The protection attribute ACCESS is implicitly set to WRITE (explicitly only with MODIFY-JV-ATTRIBUTES).

For temporary JVs, only the default values are permitted. They are not shareable, can always be overwritten and cannot be protected by passwords. The protection functions basic ACL and guards are likewise not supported. Since only the creating job can access temporary JVs up to LOGOFF processing, no protection is required against foreign access.

If more than one access control mechanism is specified for a JV, the strongest mechanism activated applies. The following table shows the method used for access control, the protection attributes, and the job variable protection hierarchy (protection levels):

Access control method	Protection attribute	Prot. level
Standard access control	ACCESS and USER-ACCESS	0
Basic access control list	BASIC-ACL	1
Access control via guards	GUARDS and SECOS	2

Table 43: Hierarchy of access control methods

All other protection attributes of a JV (e.g. passwords) are evaluated independently of the protection level.

PROTECTION-ATTR = *BY-DEF-PROT-OR-STD / *STD

Specifies from where the protection attributes with the value *BY-PROTECTION-ATTR are to be obtained.

For the assignments, see [table "Effects of PROTECTION-ATTR on protection attributes in the case of CREATE-JV" on page 2-469](#).

PROTECTION-ATTR = *BY-DEF-PROT-OR-STD

The values supplied by default protection are entered for the operands with *BY-PROTECTION-ATTR. If default protection is not active, system default values will be assigned to the relevant operands.

PROTECTION-ATTR = *STD

System default values are set for operands with *BY-PROTECTION-ATTR.

ACCESS = *BY-PROTECTION-ATTR / *WRITE / *READ

Specifies the type of accesses allowed when only standard access control is active (i.e. when there is no BASIC-ACL entry and no protection with GUARDS).

ACCESS = *BY-PROTECTION-ATTR

Standard access control is independent of the value of the PROTECTION-ATTR operand.

ACCESS = *WRITE

Standard access control allows both read and write access.

ACCESS = *READ

Standard access control allows read access only.

USER-ACCESS = *BY-PROTECTION-ATTR / *OWNER-ONLY / *ALL-USERS

Specifies whether the accesses allowed with the ACCESS operand are also available to other user IDs when only standard access control is active (i.e. when there is no BASIC-ACL entry and no protection with GUARDS).

USER-ACCESS = *BY-PROTECTION-ATTR

Access is independent of the value of the PROTECTION-ATTR operand.

USER-ACCESS = *OWNER-ONLY

Only the owner (user ID under which the JV is cataloged) and the system administration have access.

USER-ACCESS = *ALL-USERS

All user IDs have access rights.

BASIC-ACL = *BY-PROTECTION-ATTR / *NONE / STD / *PARAMETERS(...)

Specifies whether a BASIC-ACL entry is to be generated for the JV and whether access control is consequently to be performed through that entry.

BASIC-ACL = *BY-PROTECTION-ATTR

Access control via BASIC-ACL is independent of the value of the PROTECTION-ATTR operand.

BASIC-ACL = *NONE

BASIC-ACL is not activated for the JV. Access control (standard access control) is effected in accordance with the protection attributes USER-ACCESS and ACCESS (see output fields *USER-ACC* and *ACCESS* of the SHOW-JV-ATTRIBUTES command).

BASIC-ACL = *STD

A BASIC-ACL entry with the following values is created for the JV:
OWNER = *PARAMETERS(READ = *YES, WRITE = *YES),
GROUP = *NO-ACCESS and OTHERS = *NO-ACCESS

BASIC-ACL = PARAMETERS(...)

A BASIC-ACL entry is created for the JV and access control is then effected via the basic access control list (BACL).

The read and write access rights can be explicitly set or denied for each user class.

User classes are:

- OWNER, i.e. user ID of the owner and systems support
- GROUP, i.e. all user IDs which belong to the group of the owner (except the owner and systems support). Definition of user groups is possible only when the software product SECOS is used.
With regard to the possible use of SECOS, the same rights should be allocated for GROUP as for OTHERS.
- OTHERS, i.e. all user IDs which do not belong to the group of the owner.

OWNER = *NO-ACCESS / *PARAMETERS(...)

Specifies which access rights are to be set for the owner. NO-ACCESS is the default value, i.e. the owner has neither read nor write authorization.

OWNER = *PARAMETERS(...)

The owner's access rights are entered as specified:

READ = *NO / *YES

Specifies whether read authorization is set.

WRITE = *NO / *YES

Specifies whether write authorization is specified.

Write authorization does *not* imply read authorization.

GROUP = *NO-ACCESS / *PARAMETERS(...)

Specifies which access rights are to be set for all user IDs from the group of the owner.

NO-ACCESS is the default value, i.e. the user class GROUP has neither read nor write authorization.

GROUP = *PARAMETERS(...)

Access rights are to be set as specified:

READ = *NO / *YES

Specifies whether read authorization is set.

WRITE = *NO / *YES

Specifies whether write authorization is specified.

Write authorization does *not* imply read authorization.

OTHERS = *NO-ACCESS / *PARAMETERS(...)

Specifies which access rights are to be set for user IDs which do not belong to the group of the owner. If SECOS is not used, access rights should be set as for GROUP with regard to an analysis for future use of SECOS.

*NO-ACCESS is the default value, i.e. the user class OTHERS has neither read nor write authorization.

OTHERS = *PARAMETERS(...)

Access rights are to be set as specified:

READ = *NO / *YES

Specifies whether read authorization is set.

WRITE = *NO / *YES

Specifies whether write authorization is specified.

GUARDS = *BY-PROTECTION-ATTR / *NONE / *PARAMETERS(...)

Specifies whether access control is to be performed for the JV using GUARDS.

GUARDS = *BY-PROTECTION-ATTR

Access control using GUARDS is independent of the value of the PROTECTION-ATTR operand.

GUARDS = *NONE

Access to the JV is not to be controlled via GUARDS.

GUARDS = *PARAMETERS(...)

Access to the JV is to be controlled via GUARDS.

Access to the job variable is controlled via a guard, i.e. a specific object identifying all the conditions subject to which access will be granted: such as date, time, period of time, user ID. The GUARDS function unit of the chargeable software product SECOS (see the "SECOS" manual [35]) must be installed in order to create and maintain a guard.

Each guard is uniquely identified by its name. Guard names resemble JV names: they are made up of two parts, the user ID (optional) and the name part (up to 8 characters). If no user ID is specified explicitly, the user's own ID is added implicitly.

Each access mode can be controlled by a separate guard. If no guard is assigned for an access mode (*NONE), access control will refuse any corresponding access (e.g. WRITE=*NONE prevents all write access).

Specifying GUARDS=*PARAMETERS defines access control via GUARDS with all access modes being set to the default value *NONE, i.e. neither read access to the JV nor write or execute access is allowed.

The GUARDS subsystem is not required in order to define access control via GUARDS. A check by GUARDS takes place only when JV access occurs:

If a defined guard is not accessible, the mode of access protected by it is not permitted. No access at all is possible if the GUARDS subsystem is not available at the time of access.

READ = *NONE / <filename 1..18 without-cat-gen-vers>

Name of a guard controlling read access (up to 8 characters if no user ID is specified).

The default value is *NONE, i.e. no read access is granted.

WRITE = *NONE / <filename 1..18 without-cat-gen-vers>

Name of a guard controlling write access (up to 8 characters if no user ID is specified).

The default value is *NONE, i.e. no write access is granted.

READ-PASSWORD = *BY-PROTECTION-ATTR / *NONE / *SECRET / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>

Password for protection against unauthorized reading. The READ-PASSWORD operand has the following special characteristics:

- The input field is automatically blanked out in the guided dialog.
- In unguided dialog and foreground procedures, the entry *SECRET or ^, SDF provides a blanked out input field for inputting the password .
- The password entered is not logged.

READ-PASSWORD = *BY-PROTECTION-ATTR

Allocation of a read password is independent of the value of the PROTECTION-ATTR operand.

WRITE-PASSWORD = *BY-PROTECTION-ATTR / *NONE / *SECRET / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>

Write or read password for the JV to be modified. The WRITE-PASSWORD operand has the following special characteristics:

- The input field is automatically blanked out in the guided dialog.
- In unguided dialog and foreground procedures, the entry *SECRET or ^, SDF provides a blanked out input field for inputting the password .
- The password entered is not logged.

WRITE-PASSWORD = *BY-PROTECTION-ATTR

Allocation of a write password is independent of the value of the PROTECTION-ATTR operand.

MANAGEMENT-CLASS = *NONE / <composed-name 1..8>*Only for job variables on SM pubsets*

Specifies whether the HSMS functions JV backup and (long-term) archival are to be controlled via a management class defined via HSMS. See the "HSMS" manual [18] for further details.

Assignment of a management class is rejected in the following cases:

- the JV is to be created on an SF pubset
- the specified management class has not been defined for the SM pubset

SUPPRESS-ERRORS = *NONE / *JV-EXISTING

Specifies whether there is an error if the specified JV exists.

SUPPRESS-ERRORS = *NONE

If the specified JV exists, the command is rejected with an error. The error triggers the spin-off mechanism resp. the SDF-P error handling.

SUPPRESS-ERRORS = *JV-EXISTING

If the specified JV already exists, the command has no effect on the JV. There is no error (error JVS0444 is suppressed).

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
2	0	CMD0001	Command executed with a warning
	1	CMD0202	Syntax error
	32	CMD0221	System error
	64	JVS04E0	Command not executable in the call environment; if possible, remove cause of error (see SYSOUT message JVS04xx)
	130	JVS04E1	Command cannot be executed at this time; for cause see SYSOUT message JVS04xx
	130	CMD2282	Subsystem JV not available for indefinite time

Examples

```
/create-jv jv=jv.perm.error,prot=*par(user-access=*all-users,  
write-pass=c'fehl')
```

```
/show-jv-attr jv=jv.perm.error,inf=*all-attr  
%0000000 :LEO:$USER1.JV.PERM.ERROR  
% USER-ACC = ALL-USERS ACCESS = WRITE  
% CRE-DATE = 2012-03-15 EXPIR-DATE = 2012-03-15  
% CRE-TIME = 09:09:00 EXPIR-TIME = 00:00:00  
% READ-PASS = NONE  
% WRITE-PASS = YES  
%SUM 000001 JV'S; JV-VALUE = 00000000 BYTES
```

The job variable JV.PERM.ERROR is created, i.e. its name is entered in the catalog. The job variable is defined as shareable and protected by a write password.

```
/create-jv jv=jv.perm.error.read,  
prot=(basic-acl=(owner=(read=y,write=y),group=(read=y),others=(read=y)))
```

```
/show-jv-attr jv=jv.perm.error.read,inf=*all-attr  
%0000000 :LEO:$TSOS.JV.PERM.ERROR.READ  
% USER-ACC = OWNER-ONLY ACCESS = WRITE  
% OWNER = R W GROUP = R - OTHERS = R -  
% CRE-DATE = 2012-01-19 EXPIR-DATE = 2012-01-19  
% CRE-TIME = 18:18:29 EXPIR-TIME = 00:00:00  
% READ-PASS = NONE  
% WRITE-PASS = NONE  
%SUM 000001 JV'S; JV-VALUE = 00000000 BYTES
```

The job variable JV.PERM.ERROR.READ is created. It is not shareable, but read authorization is granted to other users via BASIC-ACL. The owner has both read and write access.

```
/create-jv jv=#jv.temp.work
```

```
/show-jv-attr jv=#jv.temp.work,inf=*all-attr  
%0000000 :LEO:$TSOS.S.123.4HM7.JV.TEMP.WORK  
% USER-ACC = OWNER-ONLY ACCESS = WRITE  
% CRE-DATE = 2012-01-19 EXPIR-DATE = 2012-01-19  
% CRE-TIME = 18:22:44 EXPIR-TIME = 00:00:00  
% READ-PASS = NONE  
% WRITE-PASS = NONE  
%SUM 000001 JV'S; JV-VALUE = 00000000 BYTES
```

The job variable #JV.TEMP.WORK is created temporarily. The protection attributes are set to their default values (different values cannot be specified). Only the creating job has access.

CREATE-OPERATOR-ROLE

Define name and routing codes for new operator role

Description status:	SRPMNUC V19.0A
Functional area:	Operator function control
Domain:	SECURITY-ADMINISTRATION
Privileges:	SECURITY-ADMINISTRATION

Function

This command is used to compile a list of routing (authorization) codes, to form an “operator role”, and to inform the specified pubset of it. An operator role corresponds to an area of work, and gives the exerciser of the role the right to issue the commands assigned to that area of work. Each area of work (= operator role) is represented by a set of authorization (routing) codes specified by the system support staff; this may be any combination of the total of 40 authorization codes used in BS2000.

The creation and issuing of operator roles provides a means of increasing protection against unauthorized access to the operating functions. All dynamic authorization names, i.e. operator identifications, (see “Simplifying system operation” in the “Introduction to System Administration” [14]) are treated like user IDs; for each operator identification there is an entry in the user catalog, which uniquely identifies it. If the application wishes to undertake a system operating task, it must specify an operator identification when it is connected, and then apply to be assigned an operator role, using the command REQUEST-OPERATOR-ROLE. Not until this role is assigned can it function properly as a console. The system support staff creates the link between the operator identification and the operator roles which the application may accept under this operator identification, using the MODIFY-OPERATOR-ATTRIBUTES command.

Operator roles cannot be used for physical consoles and virtual consoles with generated authorization names.

The main operating terminal, and therefore a human operator, cannot be covered by these roles.

Format

CREATE-OPERATOR-ROLE

```
OPERATOR-ROLE = <name 1..8>
,PUBSET = *HOME / <cat-id 1..4>
,ROUTING-CODES = *NONE / *ALL / list-poss(40): * / <alphanum-name 1..1>
```

Operands

OPERATOR-ROLE = <name 1..8>

Defines the name of the operator role. This name must be specified by the authorized user programs as soon as they wish to perform the tasks (=routing codes) associated with this role.

PUBSET =

Specifies the pubset into whose catalog the role is to be entered.

PUBSET = *HOME

The operator role is to be entered in the user catalog for the home pubset.

PUBSET = <cat-id 1..4>

Exact specification of the pubset into whose catalog the new operator role is to be entered.

ROUTING-CODES =

Specifies the routing codes, and hence the area of work, which are assigned to the new operator role.

ROUTING-CODES = *NONE

Default value: no routing codes will be explicitly assigned to the new operator role. The authorized user program may nevertheless accept the role, as this command will have entered it in the user catalog for the specified pubset; however, the requesting application cannot undertake any tasks.

ROUTING-CODES = *ALL

All the routing codes known to the system should be assigned to the role.

Note

A table of routing codes will be found in the "Introduction to System Administration" [14].

ROUTING-CODES = list-poss(40): * / <alphanum-name 1..1>

Detailed specification of up to 40 routing codes to be assigned to the new operator role.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
2	0	SRM6001	Command executed with a warning
	32	SRM6020	System error during command processing
	64	SRM6040	Semantic error
	130	SRM6030	Command cannot temporarily be executed

CREATE-PAGING-FILE

Create paging file on disk

Description status:	BS2000 OSD/BC V10.0A
Functional area:	System control and optimization
Domain:	SYSTEM-MANAGEMENT SYSTEM-TUNING
Privileges:	OPERATING TSOS
Routing code:	R

Function

This command is reserved for systems support staff. It creates a paging file of a specified size in contiguous blocks on a hard disk.

Format

CREATE-PAGING-FILE

VOLUME = list-poss(256): <vsn 1..6>

,**SIZE** = <integer 1..32768 *Mbyte*>

Operands

VOLUME = list-poss(256): <vsn 1..6>

Identifies by volume serial number the disk(s) on which one or more paging files of the specified size are to be set up.

Up to 256 disks can be selected.

SIZE = <integer 1..32768 *Mbyte*>

Specifies the size in megabytes of the paging file which is to be created.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
1	0	CMD0001	Command executed without error
	0	CMD0001	Warning: There is already a paging file of the specified size on the disk Guaranteed message: EMM2811
	32	EMM2800	The function cannot be implemented owing to an internal error Guaranteed messages: EMM2818, EMM2828
	64	EMM2802	The specified disk is unknown; it cannot be identified in the given configuration Guaranteed message: EMM2817
	64	EMM2803	Paging file creation not permitted for specified pubset or disk Guaranteed messages: EMM2812, EMM2813
	64	EMM2805	Operation not permitted: There is already a paging file of a different size on the disk, there is not enough free space, the pubset is not imported, the SM pubset is not fully imported. Guaranteed messages: EMM2811, EMM2814, EMM2815, EMM2825, EMM2838
	128	EMM2807	The function cannot be implemented because there are not enough resources available Guaranteed messages: EMM2819, EMM2829

Note

Command processing is aborted in the event of an error with a return code of EMM2800 or EMM2807 occurring while a list is being specified. In all other cases processing continues.

Notes

Before issuing the command, make sure that the space required for the paging file is available on the disk. The blocks should if possible all be contiguous. It may be possible to make enough contiguous space by repositioning files on the disk.

The pubset the disk belongs to must have been imported. It must not be a shared pubset, or a pubset emulated in GS.

The paging file is not backed up by ARCHIVE or HSMS.

Example

Creating a 32-megabyte paging file on a disk with the volume serial number 2OSW.0:

```
/CREATE-PAGING-FILE VOLUME=2OSW.0,SIZE=32
```


CREATE-SNAPSET

Create a Snapset

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT SNAPSET
Privileges:	TSOS HSMS-ADMINISTRATION

Function

The CREATE-SNAPSET command creates a Snapset for an imported pubset. This also includes entering the Snapset in the Snapset catalog and placing it in service. The DELETE-EARLIEST operand controls whether a new Snapset is created and under which conditions the oldest existing Snapset is to be deleted and reused before the Snapset is created (the default is deletion when the Snapset limit is reached).

The command is rejected when a new Snapset is to be created and no free Snapset is available (the setting DELETE-EARLIEST=*NO prevents the oldest Snapset from being deleted when the Snapset limit is reached).

Format

CREATE-SNAPSET

PUBSET = <cat-id 1..4>

,DELETE-EARLIEST = *AT-LIMIT / *YES / *NO

Operands

PUBSET = <cat-id 1..4>

Catalog ID of the pubset for which the Snapset is to be created.

DELETE-EARLIEST = *AT-LIMIT / *YES / *NO

Specifies whether and under which conditions the oldest Snapset is to be deleted.

DELETE-EARLIEST = *AT-LIMIT

The oldest Snapset is deleted only if the maximum number of Snapsets permissible for the pubset (Snapset limit) has already been reached.

The Snapset limit is entered in the SVL of the pubres/volres of the Control Volume Set using the SET-PUBSET-ATTRIBUTES command.

DELETE-EARLIEST = *YES

The oldest Snapset is always deleted.

DELETE-EARLIEST = *NO

A new Snapset is created. If the Snapset limit has already been reached, the command is rejected.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
	32	CMD0216	No authorization
	64	DMS1351	Internal error
	64	DMS1386	Error in the memory request
	64	DMS1389	Error in MSCF communication
	64	DMS138B	No MRSCAT entry
	64	DMS138C	Pubset not accessible
	64	DMS13D6	Snapset limit exceeded
	64	DMS148F	GCF not loaded
1	64	DMS13D7	Internal error in Snapset management: Return code of GCF
2	64	DMS13D7	Internal error in Snapset management: Return code in the case of mount/dismount
3	64	DMS13D7	Internal error in Snapset management: Return code when reading or writing the SVL
4	64	DMS13D7	Internal error in Snapset management: Return code when setting/resetting the reconfiguration lock
5	64	DMS13D7	Internal error in Snapset management: Return code of DPSVLST (create the volume list)
6	64	DMS13D7	Internal error in Snapset management: Return code of SHC-OSD
7	64	DMS13D7	Internal error in Snapset management: Return code of CCOPY
	64	DMS13DF	SHC-OSD subsystem not available

CREATE-STORAGE-CLASS

Define storage class for SM pubset

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS

Function

The CREATE-STORAGE-CLASS command allows systems support to define a new storage class for an SM pubset. The SM pubset must have been imported to the local system (in exclusive or shared mode).

An entry for the new storage class is added to the current storage class catalog (:<sm-pubset-catid>:\$TSOS.SYSCAT.STORCLS). The storage class definition contains a list of file attributes. If it is assigned a volume set list, it also includes the name of the list. The user can assign a storage class to a file in the CREATE-FILE and MODIFY-FILE-ATTRIBUTES commands. The file is then implicitly given all the file attributes defined in the storage class. If there is a volume set list assigned to the storage class, the file will if possible be stored in a volume set from the assigned volume set list.

The SHOW-STORAGE-CLASS command can be used to list the attributes of a storage class.

A storage class can be modified with the MODIFY-STORAGE-CLASS command, and its entry can be removed from the storage class catalog with the DELETE-STORAGE-CLASS command.

Systems support can use the CHANGE-STORAGE-CLASS-CATALOG command to change or, in the event of an error, dynamically load the storage class catalog.

Format

CREATE-STORAGE-CLASS
<pre> STORAGE-CLASS-NAME = <composed-name 1..8> , PUBSET = <cat-id 1..4> , FILE-ATTRIBUTES = <u>*STD</u> / *PARAMETERS(...) *PARAMETERS(...) IO-ATTRIBUTES = <u>*STD</u> / *PARAMETERS(...) *PARAMETERS(...) PERFORMANCE = *STD / *HIGH / *VERY-HIGH , USAGE = *READ-WRITE / *WRITE / *READ , DISK-WRITE = *STD / *IMMEDIATE / *BY-CLOSE , AVAILABILITY = *STD / *HIGH , FILE-PREFORMAT = *BY-PUBSET-DEFAULT / *K / *NK2 / *NK4 , WORK-FILE = *NO / *YES , STORAGE-CLASS-INFO = <u>*NONE</u> / <c-string 1..720 with-low> , PROTECTION = <u>*NONE</u> / *BY-GUARDS(...) *BY-GUARDS(...) GUARD-NAME = <filename 1..18 without-cat-gen-vers> , VOLUME-SET-LIST = <u>*NONE</u> / <composed-name 1..8> </pre>

Operands

STORAGE-CLASS-NAME = <composed-name 1..8>

Defines a name for the new storage class.

PUBSET = <cat-id 1..4>

Specifies the catalog ID of the pubset for which the storage class is being created. The pubset must be an SM pubset which has been locally imported (in exclusive or shared mode).

FILE-ATTRIBUTES = *STD / *PARAMETERS(...)

Specifies pubset management requirements which can be represented in the form of file attributes. The file attributes apply to all files to which the user assigns the storage class instead of explicit attributes (see the STORAGE-CLASS operand of the CREATE-FILE or MODIFY-FILE-ATTRIBUTES command).

FILE-ATTRIBUTES = *STD

Files assigned this storage class are allocated standard pubset management requirements. These correspond to the default values in the FILE-ATTRIBUTES=*PARAMETERS(...) structure.

FILE-ATTRIBUTES = *PARAMETERS(...)

Specifies which file attributes are allocated to files in this storage class.

IO-ATTRIBUTES = *STD / *PARAMETERS(...)

Specifies which performance requirements apply to files in this storage class.

IO-ATTRIBUTES = *STD

The default values in the *PARAMETERS(...) structure apply.

IO-ATTRIBUTES = *PARAMETERS(...)

Performance attributes are determined by the specifications in the PERFORMANCE and USAGE operands.

PERFORMANCE = *STD / *HIGH / *VERY-HIGH

Specifies the performance attribute for files in this storage class. It indicates the priority required for the I/O operations selected in the USAGE operand.

PERFORMANCE = *STD

No performance requirements are defined for files in this storage class. In this case the USAGE operand has no impact on the processing of the file.

PERFORMANCE = *HIGH

A cache should be used to process files in this storage class (high performance priority).

PERFORMANCE = *VERY-HIGH

A cache should be used to process files in this storage class, and if possible the entire file should be kept permanently in the cache (highest performance priority, only available in the GS cache medium).

USAGE = *READ-WRITE / *WRITE / *READ

Specifies the I/O operations for which enhanced performance (caching) is required. The default is *READ-WRITE, i.e. the requirement applies to both read and write operations.

The requirement may also be applied only to write operations (*WRITE) or only to read operations (*READ).

If the file has no special performance attribute (PERFORMANCE=*STD), the USAGE operand has no impact on processing.

DISK-WRITE = *STD / *IMMEDIATE / *BY-CLOSE

Specifies the time at which data consistency is required for files in this storage class after a write operation.

DISK-WRITE = *STD

The default setting of *STD is equivalent to *IMMEDIATE for permanent files and to *BY-CLOSE for temporary files.

DISK-WRITE = *IMMEDIATE

The data in a file must be in a consistent state immediately on completion of a write operation, so a volatile write cache should not be used to process files in this storage class.

DISK-WRITE = *BY-CLOSE

The data in a file does not have to be in a consistent state until after CLOSE processing, so a volatile write cache may be used to process files in this storage class.

AVAILABILITY = *STD / *HIGH

Specifies availability requirements for files in this storage class.

AVAILABILITY = *STD

There are no special availability requirements.

AVAILABILITY = *HIGH

Files in this storage class are required to have high availability (e.g. storage on mirrored disks).

FILE-PREFORMAT = *BY-PUBSET-DEFAULT / *K / *NK2 / *NK4

Governs the preferred file format for files in this storage class. This specification only applies when the catalog entry is created; it is used to automatically determine the storage location.

As long as the file does not yet have a creation date (CRE-DATE=NONE), the storage location is provisional and may still change in the course of OPEN processing.

FILE-PREFORMAT = *BY-PUBSET-DEFAULT

The pubset-global default applies.

FILE-PREFORMAT = *K

File in this storage class are to be created as K files.

FILE-PREFORMAT = *NK2

File in this storage class are to be created as NK2 files.

FILE-PREFORMAT = *NK4

File in this storage class are to be created as NK4 files.

WORK-FILE = *NO / *YES

Governs whether files in this storage class are work files.

WORK-FILE = *NO

Files in this storage class are created as normal files.

WORK-FILE = *YES

Files in this storage class are created as work files. Systems support can delete work files at appointed times.

STORAGE-CLASS-INFO = *NONE / <c-string 1..720 with-low>

Systems support can compose a brief text describing storage class attributes. It will typically relate to special subset management requirements which are defined by systems support but cannot be represented in the form of file attributes. This brief text is included in the output of the SHOW-STORAGE-CLASS command to inform the user of any special properties of the storage class.

STORAGE-CLASS-INFO = *NONE

There is to be no explanatory text for the storage class.

STORAGE-CLASS-INFO = <c-string 1..720 with-low>

A brief text describing special properties of the storage class as defined by systems support.

PROTECTION = *NONE / *BY-GUARDS(...)

Specifies whether access to the storage class is to be controlled by guards.

PROTECTION = *NONE

Access to the storage class is not to be controlled by guards.

PROTECTION = *BY-GUARDS(...)

Allocation of the storage class to a file and the listing of the storage class definitions are to be controlled by a guard. The guard named next defines conditions under which a user is allowed to access the storage class.

GUARD-NAME = <filename 1..18 without-cat-gen-vers>

Name of the guard.

VOLUME-SET-LIST = *NONE / <composed-name 1..8>

Specifies whether a volume set list is to be assigned to the storage class.

The default is *NONE, i.e. no volume set list is to be assigned to the storage class at this time.

VOLUME-SET-LIST = <composed-name 1..8>

The specified volume set list is assigned to the storage class. The volume set list must already exist for this SM subset.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
	1	CMD0202	Syntax error in command
	32	CMD0221	Internal system error
	64	CMD0216	No authorization to issue command
	64	DMS1488	Storage class already defined
	64	DMS1497	Invalid file attribute combination
	64	DMS148B	Volume set list not defined
	64	DMS1485	Pubset not known
	64	DMS1486	Pubset is not a system-managed pubset
	64	DMS1487	Pubset not available
	64	DMS1490	Storage class management not available for this pubset
	64	DMS1483	Storage class catalog invalid
	64	DMS1481	Error accessing storage class catalog
	64	DMS1484	Volume set list catalog invalid
	64	DMS1482	Error accessing volume set list catalog
	129	DMS148D	Not enough class 4/5 memory
	129	DMS148E	Error on MSCF connection to master
	129	DMS148F	GCF subsystem not ready

CREATE-TAPE-SET

Generate volume serial number set (tape set)

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING
Routing code:	\$ (with NBCONOPI=N) or E (with NBCONOPI=Y)

Function

The CREATE-TAPE-SET command is used to create a set of VSNs which can be used for producing tape files. The link with the tape files which are to be processed is set up by the ADD-FILE-LINK command (operand TAPE-SET-NAME). The set of VSNs is entered in the TST (TAPE SET TABLE). Within the TST, the TAPE-SET-NAME performs the same function as does the file link name within the TFT.

Information on TST entries can be obtained with the SHOW-FILE-LINK command.

Format

CREATE-TAPE-SET

```

TAPE-SET-NAME = <alphanum-name 1..4>
, VOLUME = [*ANY](...) / list-poss(255): <alphanum-name 1..6>
  [*ANY](...)
  |   NUMBER-OF-DEVICES = 1 / <integer 1..9>
, DEVICE-TYPE = *TAPE / <device>

```

Operands

TAPE-SET-NAME = <alphanum-name 1..4>

The name of the tape set; this establishes the link to the TST with the specified name, or creates a new TST.



The tape set name must not be used simultaneously within the same task as a file link name.

VOLUME = *ANY(...) / list-poss(255): <alphanum-name 1..6>in

VSNs of the tapes which are to be allocated to the tape set.

VOLUME = *ANY(...)

The operator, or MAREN if available, should mount any suitable tapes on the specified device type (no particular VSN is required).

NUMBER-OF-DEVICES = 1 / <integer 1..9>

The number of tapes which is required.

DEVICE-TYPE = *TAPE / <structured-name 1..8>

The device type to which the tapes are assigned.

The value to be specified is the device type or volume type of the tapes, from which the Device Management System determines the device type which is to be made available for tape processing.

Only device types or volume types known in the system are accepted. In interactive mode, DEVICE-TYPE=? calls up a list of the available device and volume types.

The default value is TAPE, i.e. for tape processing the devices to be used should support a recording density of 1600 or 6250 bpi (bytes per inch). Other permitted specifications will be found in the table in [section "Device types for DMS tape processing" on page 1-84](#).

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed
	1	CMD0202	Syntactical or semantic error in command
	32	DMS0584	A state that does not allow the function to continue was reported during processing
	32	DMS05C7	Unexpected internal error in DMS
	64	DMS0586	It is not possible to access or reserve a volume at present
	64	DMS0587	Use of the specified command has been restricted by the system administrator
	64	DMS06FF	BCAM connection severed
	130	DMS0524	System address space exhausted
	130	DMS0586	It is not possible to access or reserve a volume at present
	130	DMS0594	Not enough virtual memory available

Example*Creation of a tape set with subsequent file processing*

```

/cre-tape-set tape-set-name=tap1,vol=(d2312k,d2315k,d2322k),
dev-type=tape-c6 _____ (1)
/show-file-link link=tap1,inf=(vol=*yes) _____ (2)
%
%-- LINK-NAME ----- FILE-NAME -----
% T TAP1 _____ *DUMMY _____
% ----- VOLUME -----
% NUM-OF-VOL = 3          NUM-OF-DEV = 0
% DEV-TYPE   = TAPE-C6    T-SET-NAME = TAP1
% T-SET-SHR  = 1          F-SET-ID   = D2312K
% VSN/DEV    = D2312K     D2315K     D2322K
% T-SET-VSN  = (D2312K)  D2315K     D2322K
/cre-file max.tape-file.3,sup=*tape(vol=*no,dev-type=tape-c6,premount=0) _____ (3)

/add-file-link link=dmcopy22,file-name=max.tape-file.3,
number-of-premounts=0
sup=*tape(vol-list=*tape-set(tap1)) _____ (4)

/show-file-link link=dmcopy22,inf=(file-contr=*yes,vol=*yes) _____ (5)
%
%-- LINK-NAME ----- FILE-NAME -----
% T DMCOPY22 _____ :20S2:$USER1.MAX.TAPE-FILE.3 _____
% ----- FILE-CONTROL-BLOCK - GENERAL -----
% ACC-METH   = *BY-PROG   OPEN-MODE   = *BY-PROG   REC-FORM    = *BY-PROG
% REC-SIZE   = *BY-PROG   BUF-LEN    = *BY-PROG   BLK-CONTR   = *BY-PROG
% F-CL-MSG   = STD       CLOSE-MODE  = *BY-PROG
% ----- FILE-CONTROL-BLOCK - DISK -----
% SHARED-UPD = *BY-PROG   WR-CHECK   = *BY-PROG   IO(PERF)    = *BY-PROG
% IO(USAGE)  = *BY-PROG   LOCK-ENV   = *BY-PROG
% ----- FILE-CONTROL-BLOCK - TAPE -----
% LABEL      = *BY-PROG   (DIN-R-NUM = *BY-PROG, TAPE-MARK   = *BY-PROG)
% CODE       = *BY-PROG   EBCDIC-TR  = *BY-PROG   F-SEQ       = *BY-PROG
% CP-AT-BLIM = *BY-PROG   CP-AT-FEOV = *BY-PROG   BLOCK-LIM   = *BY-PROG
% REST-USAGE = *BY-PROG   BLOCK-OFF  = *BY-PROG   TAPE-WRITE  = *BY-PROG
% STREAM     = *BY-PROG
% ----- FILE-CONTROL-BLOCK - ISAM -----
% KEY-POS    = *BY-PROG   KEY-LEN    = *BY-PROG   POOL-LINK   = *BY-PROG
% LOGIC-FLAG = *BY-PROG   VAL-FLAG   = *BY-PROG   PROPA-VAL   = *BY-PROG
% DUP-KEY    = *BY-PROG   PAD-FACT   = *BY-PROG   READ-I-ADV  = *BY-PROG
% WR-IMMED   = *BY-PROG   POOL-SIZE  = *BY-PROG
% ----- VOLUME -----
% NUM-OF-VOL = 3          NUM-OF-DEV = 1
% DEV-TYPE   = *NONE     T-SET-NAME = TAP1
% T-SET-SHR  = 2          F-SET-ID   = D2312K
% VSN/DEV    = D2312K     D2315K     D2322K
% T-SET-VSN  = (D2312K)  D2315K     D2322K
/copy-file from=1st.bsp.2,to=max.tape-file.3 _____ (6)
/show-file-attr max.tape-file.3,inf=(alloc=*yes) _____ (7)
%
%----- ALLOCATION -----
% SUPPORT    = PVT
% EXTENTS    VOLUME      DEVICE-TYPE  EXTENTS    VOLUME      DEVICE-TYPE
%            D2312K     TAPE-C6     (          D2315K     TAPE-C6 )
% (          D2322K     TAPE-C6 )
%:20S2: TAPE : 1 FILE

```

- (1) Creates the tape set *TAPI* containing the three magnetic tape cartridges (volume type TAPE-C6) with the volume IDs *D2312K*, *D2315K*, and *D2322K*.
- (2) Output of the TST entry with the command SHOW-FILE-LINK.
- (3) Creates a catalog entry for a tape file under the file name *MAX.TAPE.FILE.3*. No volume identifier has been defined as yet: TAPE-C6 is specified as the volume type. It is not necessary to reserve a magnetic tape cartridge in this case (NUMBER-OF-PREMOUNTS=0).
- (4) A TFT under the link name *DMCOPY22* is created for the file *MAX.TAPE.FILE*, since the COPY-FILE command is to be used to copy data into this file. At the same time, the TFT entry is linked with the tape set *TAPI* (VOLUME-LIST= *BY-TAPE-SET).
- (5) Output of the TFT entry with the link name *DMCOPY22*. The output includes relevant information on file processing and volumes.
- (6) The contents of file *LST.BSP.2* are copied to the file *MAX.FILE.3*.
- (7) Returns information from the catalog entry of the file *MAX.FILE.3* showing volume allocation attributes. The file was created on the magnetic tape cartridge *D2312K*. In this case, however, the volume IDs *D2315K* and *D2322K* (from the TST entry) are also entered into the volume list of the catalog entry. These volumes are to be used if the file *MAX.TAPE.3* is extended and the tape cartridge *D2312K* does not have enough space for this purpose.

CREATE-VOLUME-SET-LIST

Define volume set list for SM pubset

Description status:	BS2000 OSD/BC V10.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS

Function

The CREATE-VOLUME-SET-LIST command allows systems support to define a new volume set list for a system-managed pubset (SM pubset). The SM pubset must have been imported to the local system (in exclusive or shared mode).

An entry for the new volume set list is added to the current volume set list catalog (:<sm-pubset-catid>:\$TSOS.SYSCAT.VSETLST). Up to 255 volume sets can be assigned to a volume set list. If the volume set list is associated with a storage class (see the VOLUME-SET-LIST operand in the CREATE- or MODIFY-STORAGE-CLASS command), files in that storage class will if possible be stored in volume sets from the assigned volume set list

The SHOW-VOLUME-SET-LIST command provides information about the composition of volume set lists.

A volume set list can have volume sets added to or removed from it with the MODIFY-VOLUME-SET-LIST command, and its entry can be removed from the volume set list catalog with the DELETE-VOLUME-SET-LIST command.

Systems support can use the CHANGE-VOLUME-SET-LIST-CATALOG command to change or, in the event of an error, dynamically load the volume set list catalog.

Format

CREATE-VOLUME-SET-LIST

```

VOLUME-SET-LIST-NAME = <composed-name 1..8>
, PUBSET = <cat-id 1..4>
, VOLUME-SET = *NONE / list-poss (255): <cat-id 1..4>
, VOLUME-SET-LIST-INFO = *NONE / <c-string 1..720 with-low>

```

Operands

VOLUME-SET-LIST-NAME = <composed-name 1..8>

Defines a name for the new volume set list.

PUBSET = <cat-id 1..4>

Specifies the catalog ID of the SM pubset for which the volume set list is being created. The pubset must have been imported to the local system (in exclusive or shared mode).

VOLUME-SET = *NONE / list-poss(255): <cat-id 1..4>

Specifies which volume sets are to be assigned to the volume set list.

VOLUME-SET = *NONE

No volume sets are to be assigned to the volume set list at this time.

VOLUME-SET = list-poss(255): <cat-id 1..4>

The specified volume sets are to be assigned to the volume set list. Up to 255 volume sets can be listed.

The command does not check that the specified volume sets belong to the SM pubset for which the volume set list is being created.

VOLUME-SET-LIST-INFO = *NONE / <c-string 1..720 with-low>

Systems support can compose a brief text describing the volume set list. It will typically relate to differences between volume set lists

The default is *NONE, i.e. there is to be no explanatory text.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No errors
	1	CMD0202	Syntax error in command
	32	CMD0221	Internal system error
	64	CMD0216	No authorization to issue command
	64	DMS1489	Volume set list already defined
	64	DMS1485	Pubset not known
	64	DMS1486	Pubset is not a system-managed pubset
	64	DMS1487	Pubset not available
	64	DMS1490	Storage class management not available for this pubset
	64	DMS1484	Volume set list catalog invalid
	64	DMS1482	Error accessing volume set list catalog
	129	DMS148D	Not enough class 4/5 memory
	129	DMS148E	Error on MSCF connection to master
	129	DMS148F	GCF subsystem not ready

DEACTIVATE-SNAPSHOT

Disable dump generator SNAP

Description status:	BS2000 OSD/BC V10.0A
Functional area:	System control and optimization
Domain:	SYSTEM-MANAGEMENT
Privileges:	TSOS OPERATING
Routing code:	R

Function

The DEACTIVATE-SNAPSHOT command disables the SNAP dump function. Any suspended SNAP dump will be processed only after the SNAP dump function is enabled again. The command can be executed multiple times, e.g. in order also to delete the SNAP system files retroactively.

DEACTIVATE-SNAPSHOT is executed asynchronously. Message NSP4000 confirms that the command has been received correctly. The SHOW-SNAPSHOT-STATUS command enables you to check the modified settings.



Manual changes to the Snapshot files can lead to undefined statuses when the SNAP dump takes place. Modifications may only be made using the ACTIVATE-SNAPSHOT and DEACTIVATE-SNAPSHOT commands.

Format

DEACTIVATE-SNAPSHOT
DELETE-FILES = <u>*NO</u> / *YES

Operands

DELETE-FILES = *NO / *YES

Specifies whether system files \$TSOS.SNAPFILE and \$TSOS.SNAPFILE.DAT are to be deleted.

DELETE-FILES = *NO

The SNAP dump function is disabled but the system files are retained.

DELETE-FILES = *YES

The SNAP dump function is disabled and the system files are deleted.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without error

DECLARE-PARAMETER

Declare procedure parameters

Description status:	SDF-P-BASYS V2.5E
Functional area:	Procedures
Domain:	PROCEDURE
Privileges:	STD-PROCESSING OPERATING HARDWARE-MAINTENANCE SECURITY-ADMINISTRATION SAT-FILE-MANAGEMENT SAT-FILE-EVALUATION

Function

The DECLARE-PARAMETER command declares procedure parameters in an S procedure together with the method used to pass the parameter values to the procedure (initial value, prompting, etc.).

It is part of the procedure head. If several DECLARE-PARAMETER commands are issued, they must be combined to form a command block (declaration section) by means of BEGIN-PARAMETER-DECLARATION and END-PARAMETER-DECLARATION commands.

Restrictions

Specification of values other than the defaults for the TYPE and TRANSFER-TYPE operands is possible only if the chargeable SDF-P subsystem has been loaded.

Format

<pre> DECLARE-PARAMETER NAME = list-poss(2000): <structured-name 1..20>(…) <structured-name 1..20>(…) INITIAL-VALUE = <u>*NONE</u> / *PROMPT(…) / <text 0..1800 with-low <i>expr</i>> *PROMPT(…) PROMPT-STRING = <u>*STD</u> / <text 0..1800 with-low <i>string-expr</i>> ,DEFAULT-VALUE = <u>*NONE</u> / <text 0..1800 with-low <i>expr</i>> ,SECRET-INPUT = <u>*NO</u> / *YES ,TYPE = <u>*ANY</u> / *STRING / *INTEGER / *BOOLEAN ,TRANSFER-TYPE = <u>*BY-VALUE</u> / BY-REFERENCE </pre>

Operands

NAME = list-poss(2000): <structured-name 1..20>(…)

Defines the names of the procedure parameters used.

Procedure parameters are S variables that are known within the procedure.

The following attributes are declared for each procedure parameter:

INITIAL-VALUE =

This defines the initial value.

INITIAL-VALUE = *NONE

The procedure parameter is not initialized, i.e. no initial value is declared. When the procedure is called a value must be assigned to the procedure parameter (see the CALL-PROCEDURE or ENTER-PROCEDURE command).

INITIAL-VALUE = *PROMPT(…)

If no value is declared when the procedure is called, SDF-P queries the value when the procedure parameter occurs for the first time (prompting). Prompting is only possible within a dialog. If prompting is not possible or if no value has been entered in the dialog then the initial value declared in the DEFAULT-VALUE operand is used.

PROMPT-STRING =

Defines a string that is output as the prompt string. The text specified for DEFAULT-VALUE = . . . is added to the prompt string. The prompt always ends with a colon. The prompt therefore has the following form:

```
<prompt-string>_(DEFAULT = <default-value>)_:
```

PROMPT-STRING = *STD

By default, the parameter name specified in NAME= . . . (variable name) is output.

PROMPT-STRING = <text 0..1800 with-low *string-expr*>

Defines the string that is to be output as the prompt string.

DEFAULT-VALUE =

Defines an initial value for cases where no input (i.e. only DUE) is entered in the dialog or the procedure runs in the background. The value is output (for information) as part of the prompt.

DEFAULT-VALUE = *NONE

No (default) string is declared.

DEFAULT-VALUE = <text 0..1800 with-low *expr*>

Expression that is used as the default for the initial value. The specified expression must match the parameter type.

SECRET-INPUT = *NO / *YES

You can specify whether the dialog input is to be protected or entered in a nondisplaying field. In this case, the input is not logged.

INITIAL-VALUE = <text 0..1800 with-low *expr*>

The procedure parameter has the initial value derived from the specified expression (for permitted expressions see the [section "SDF-P-BASYS" on page 1-131](#)). This initial value is assumed if no other value is declared when the procedure is called.

TYPE = *ANY / *STRING / *INTEGER / *BOOLEAN

This defines the variable type of the procedure parameter.

Specification of an operand value other than the default value *ANY is possible only if the chargeable subsystem SDF-P has been loaded.

TRANSFER-TYPE = *BY-VALUE / *BY-REFERENCE

This defines the type of parameter transfer.

Specification of an operand value other than the default value *BY-VALUE is possible only if the chargeable subsystem SDF-P has been loaded.

Return codes

The DECLARE-PARAMETER command can only be used within the procedure header of an S procedure. SDF-P detects errors in the procedure head during pre-analysis and terminates the procedure call. The command return codes can only occur if the command is used outside the procedure head.

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD0202	Syntax error
	1	SDP0118	Command in incorrect context
	3	CMD2203	Incorrect syntax file
	32	CMD0221	System error (internal error)
	130	SDP0099	No further address space available

DECLARE-VARIABLE

Declare as variable

Description status:	SDF-P-BASYS V2.5E
Functional area:	Procedures
Domain:	PROCEDURE
Privileges:	STD-PROCESSING OPERATING HARDWARE-MAINTENANCE SECURITY-ADMINISTRATION SAT-FILE-MANAGEMENT SAT-FILE-EVALUATION

Function

DECLARE-VARIABLE is used to create variables and set the attributes and possibly an initial value as well.

Job variables can be integrated in SDF-P via the CONTAINER operand.

Restrictions

If the chargeable SDF-P subsystem is not available, the following restrictions apply:

- Only simple S variables can be declared (TYPE=*ANY and MULTIPLE-ELEMENTS=*NO).
- Declarations can only be accepted from a variable container (CONTAINER=<composed-name>). Otherwise it is only possible to specify CONTAINER=*STD.
- The scope can be defined with SCOPE=*CURRENT(...) only. Within the *CURRENT(...) structure, only the default value IMPORT-ALLOWED=*NO is allowed.

Format

DECLARE-VARIABLE	Alias: DCV
<p>VARIABLE-NAME = list-poss(2000): <structured-name 1..20>(…)</p> <p><structured-name 1..20>(…)</p> <ul style="list-style-type: none"> INITIAL-VALUE = <u>*NONE</u> / <text 0..1800 with-low <i>expr</i>> ,TYPE = <u>*ANY</u> / *STRING / *INTEGER / *BOOLEAN / *STRUCTURE(…) *STRUCTURE(…) DEFINITION = <u>*DYNAMIC</u> / *BY-SYSCMD / <structured-name 1..20> <p>,MULTIPLE-ELEMENTS = <u>*NO</u> / *ARRAY(…) / *LIST(…)</p> <ul style="list-style-type: none"> *ARRAY(…) LOWER-BOUND = <u>0</u> / *NONE / <integer -2147483648..2147483647> , UPPER-BOUND = <u>*NONE</u> / <integer -2147483648..2147483647> *LIST(…) LIMIT = <u>*NONE</u> / <integer 1..2147483647> <p>,SCOPE = <u>*CURRENT</u>(…) / *PROCEDURE(…) / *TASK(…)</p> <ul style="list-style-type: none"> *<u>CURRENT</u>(…) IMPORT-ALLOWED = <u>*NO</u> / *YES *PROCEDURE(…) IMPORT-ALLOWED = <u>*NO</u> / *YES *TASK(…) STATE = <u>*ANY</u> / *NEW / *OLD <p>,CONTAINER = <u>*STD</u> / <composed-name 1..64> / *VARIABLE(…) / *JV(…)</p> <ul style="list-style-type: none"> *VARIABLE(…) VARIABLE-NAME = <structured-name 1..20> , SCOPE = <u>*VISIBLE</u> / *TASK *JV(…) JV-NAME = <filename 1..54> , STATE = <u>*ANY</u> / *NEW / *OLD 	

Operands

VARIABLE-NAME = list-poss (2000): <structured-name 1..20>(…)

Declares the variable name, i.e. the name of a simple variable which is not an element in a compound variable, or the name of a compound variable.

INITIAL-VALUE = *NONE

The variable is not initialized.

For a new variable, this means that the variable does not contain an initial value. A read access would produce an error.

If the variable is already present, its contents remain unchanged; it is not assigned a new initial value.

INITIAL-VALUE = <text 0..1800 with-low *expr*>

Assigns an initial value to a new variable; the value must match the data type of the variable and can also be specified as an expression.

The entry is ignored for existing variables; they are not assigned a new initial value.

Compound variables cannot be initialized in their entirety, i.e. INITIAL-VALUE cannot be used to assign a new initial value to these variables.

TYPE = *ANY / *STRING / *INTEGER / *BOOLEAN / *STRUCTURE(…)

Assigns the data type to the variable.

TYPE = *ANY

The variable can be assigned any value of data types STRING, INTEGER and BOOLEAN.



The operand values *STRING, *INTEGER, *BOOLEAN and *STRUCTURE(…) can only be specified if the chargeable SDF-P subsystem is loaded. The operand values are described in full in the “SDF-P” manual [34].

MULTIPLE-ELEMENTS = *NO / *ARRAY(…) / *LIST(…)

Specifies whether there can be multiple elements of a variable in an array or a list.

MULTIPLE-ELEMENTS = *NO

Indicates that the variable is not an array or list.



The operand values *ARRAY(…) and *LIST(…) can only be specified if the chargeable SDF-P subsystem is loaded. The operand values are described in full in the “SDF-P” manual [34].

SCOPE = *CURRENT(...) / *PROCEDURE(...) / *TASK(...)

Defines the variable scope.

SCOPE = *CURRENT(...)

The variable is a procedure-local variable.

This corresponds to the PROCEDURE argument in call procedures.

In include procedures, *CURRENT means that the variable is declared in the current include procedure. It is then visible in this include procedure and in all include procedures on lower nesting levels (= scope: include). The lower-level IMPORT-ALLOWED operand is preset to *NO (default value), which means that the declared variable cannot be imported using IMPORT-VARIABLE. If the chargeable SDF-P subsystem is not loaded, only this default value is allowed.



The operand values *PROCEDURE(...) and *TASK(...) can only be specified if the chargeable SDF-P subsystem is loaded. The operand values are described in full in the “SDF-P” manual [34].

CONTAINER = *STD / <composed-name 1..64> / *VARIABLE(...) / *JV(...)

Defines the assignment of the variable to a container.

CONTAINER = *STD

The variable is not assigned a variable container. The value of the variable is stored in class 5 memory.

CONTAINER = <composed-name 1..64>

Assigns the variable container specified here to the variable declared with this command. Only a previously opened variable container may be specified. The specification of “STD” is not permissible here because “STD” is not interpreted as a permanent variable container.



The operand values *VARIABLE(...) and *JV(...) can only be specified if the chargeable SDF-P subsystem is loaded. The operand values are described in full in the “SDF-P” manual [34].

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	No errors
1	0	CMD0001	Warning: Element already declared
	1	CMD0202	Syntax error
	3	CMD2203	Incorrect syntax file
	32	CMD0221	System error (internal error)
	64	CMD0216	User does not have required privilege
	64	SDP0091	Semantic error
	130	SDP0099	No more address space available

Examples

Example 1

The procedure-local variable DATA of type *ANY is initialized with the string 'ANTON' and subsequently, the value of the variable is output with the SHOW-VARIABLE command:

```
/declare-variable data(c'Anton',*any)
```

```
/show-variable data
```

```
DATA = Anton
```

Example 2

The variable container MY-CONT is opened. Then the S variable MY-VAR-1 is declared as a container variable. Once a value has been assigned, the variable container (and thus also the S variable) is saved. When the variable container has been closed, the S variable MY-VAR-1 can no longer be accessed:

```
/declare-variable my-var-1, container=my-cont
```

```
/open-variable-container my-cont, from=*lib-elem(lib=my-lib)
```

```
/my-var-1='saved value'
```

```
/show-variable my-var-1
```

```
MY-VAR-1 = saved value
```

```
*END-OF-CMD
```

```
/save-variable-container my-cont
```

```
/...
```

```
/close-variable-container my-cont
```

```
/show-variable
```

```
% SDP1030 CONTAINER / VARIABLE-CONTAINER 'MY-CONT' DOES NOT EXIST
```

```
% SDP0234 OPERAND 'NAME' INVALID
```

DECRYPT-FILE

Decrypt an encrypted file

Description status:	BS2000 OSD/BC V10.0A
Functional area:	File processing
Domain:	FILE
Privileges:	STD-PROCESSING TSOS SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT

Function

The DECRYPT-FILE command converts an encrypted file into an unencrypted file. This is only possible if the crypto password specified for this file in the ENCRYPT-FILE command is entered in the task-specific crypto password table (see the ADD-CRYPTO-PASSWORD command).

Format

DECRYPT-FILE
FILE-NAME = <filename 1..54 without-gen>

Operands

FILE-NAME = <filename 1..54 without-gen>

Name of the file to be converted.

Return codes

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	CMD0001	Command executed
	1	CMD0202	Syntactical or semantic error in the command
1	0	CMD0001	No action required
2	0	DMS05B6	Time conversion UTC to LT errored
2	0	DMS05F5	Defective blocks were not copied
	130	DMS0524	System address space exhausted
	130	DMS0582	The file is currently locked or in use and cannot be processed

(Part 1 of 2)

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	130	DMS0585	An error was detected during catalog processing or during multiprocessor processing.
	130	DMS0588	Disk storage could not be assigned
	130	DMS0594	Insufficient virtual memory available
	32	DMS0584	During processing a status was reported which prevents the function from being continued.
	32	DMS05C7	Unexpected internal error in DMS
	64	CMD0216	Privileges error
	64	DMS0501	Requested catalog not available
	64	DMS0512	Requested catalog not found
	64	DMS051B	Requested user ID not in pubset Guaranteed message: DMS051B
	64	DMS051C	User does not have access right for pubset Guaranteed message: DMS051C
	64	DMS0535	Specified file not shareable
	64	DMS057C	Processing not possible because of HSMS error
	64	DMS057E	File migrated, HSMS not available
	64	DMS0585	An error was detected during catalog processing or during multiprocessor processing.
	64	DMS0588	Disk storage could not be assigned
	64	DMS05FC	Specified user ID not in the home pubset
	64	DMS0609	Access to system file not possible
	64	DMS0681	DMS error while executing job
	64	DMS0684	File does not exist
	64	DMS06B5	File open or catalog entry not updated after system error

(Part 2 of 2)

