User Guide - English

FUJITSU

ARCHIVE V9.0

Edition September 2010

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 © Fujitsu Technology Solutions GmbH 2010.

All rights reserved.
Delivery subject to availability; right of technical modifications reserved.

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

Contents

1 Preface

ARCHIVE is a software product which can be used for

- saving and restoring files and job variables and
- transferring files and job variables between different BS2000/OSD systems.

ARCHIVE can be used with directory files, thereby offering the following facilities:

- management of saved data sets, catering not only for full data saving but also for incremental and partial saving,
- assignment of the save volumes from a volume pool,
- a simple restoration function in the event of data loss.

ARCHIVE V9.0 can be used in the BS2000/OSD operating system V6.0 and higher (for SQ and SX servers in the software package OSD/XC V2.0 and higher) on the current servers from Fujitsu in all hardware series and operates all backup media which have been released for BS2000/OSD V6.0 and higher.

ARCHIVE V9.0 is a part of the software product HSMS V9.0.

1.1 Target group

This manual is intended for system support and nonprivileged users of the operating system BS2000/OSD.

The reader should be familiar with the operating system BS2000/OSD-BC, in particular with the most important commands.

The following manuals will prove useful in this respect: “Commands” [4], “DMS Introductory Guide” [2] and “Job Variables” [11].

1.2 Summary of contents

The manual is divided into a number of chapters which contain information about the software product ARCHIVE and provide support for day-to-day operation.

- Chapters 2 (Introduction to ARCHIVE), 4 (Execution and environment) and 5 (ARCHIVE functions) serve as an introduction to ARCHIVE
- Chapters 6 (ARCHIVE statements) and 7 (Application examples) are designed to assist in daily operation
- Chapter 3 (Installation) is required when installing ARCHIVE
- Chapter 8 (ARCHIVE macro) is required when programming ARCHIVE calls.

You will find various indexes at the back of the manual that will make working with this manual easier.

README file

Any additions to the manuals are described in the Readme files for the various product versions. These Readme files are available at *http://manuals.ts.fujitsu.com* under the various products.

Readme file under BS2000/OSD

On your BS2000 system you will find Readme files for the installed products under the file name:

```
SYSRME.<product>.<version>.E
```

Please refer to the appropriate system administrator for the user ID under which the required Readme file can be found. You can also obtain the path name of the Readme file directly by entering the following IMON command:

```
/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>,LOGICAL-ID=SYSRME.E
```

You can view the Readme file with `/SHOW-FILE` or by opening it in an editor or print it at a standard printer using the following command (e.g. ARCHIVE V9.0):

```
/PRINT-DOCUMENT FROM-FILE=SYSRME.ARCHIVE.090.E,
                LINE-SPACING=*BY-EBCDIC-CONTROL
```

Additional product informations

Current information, version and hardware dependencies and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available at *http://manuals.ts.fujitsu.com*.

1.3 Changes since the last version of the manual

Since the previous manual, a number of major changes have been made:

- The ARCHIVE README file has been integrated.
- The manual has been brought in line with BS2000/OSD V8.0.
- The installation files have the suffix “090”.
  New installation file SKMLNK.ARCHIVE.090.TPR for SQ servers.
- Information on the ARCHIVE messages is provided in the section “ARCHIVE messages” on page 29.
- Information on saving and reconstructing encrypted files.
- Saving with tape encryption on the volume type TAPE-U4E.
- Backup files on disk are created in a neutral NK file format, regardless of the format of the private disk or pubset.
- A file on several private disks with different device types can also be saved and restored (ARCHIVE and HSMS operation).
- Further performance measures.
- The default value for the CONVERSION operand in the ARCHIVE statements IMPORT and RESTORE has been changed from NO to STD.
- When systems support performs a RESTORE, a missing user ID can be configured by means of an optional REP.
- New operand value BLOCK-SIZE=STD in the EXPORT and SAVE statements.
- Files with the BACKUP entry E can also be saved with the SAVE statement.
- In the HELP statement the information is by default output in the language set for message output.
- New BIG tape format for large tape blocks.

The chapter “Processing directory files with DIRCONV” is no longer included.
The description can be found in the chapter of the same name in the “HSMS” manual [9].

The “Messages” chapter is no longer included.
You will find the messages using an HTML application on our manual server (URL: *http://manuals.ts.fujitsu.com*) and on the DVD “BS2000/OSD SoftBooks”.

1.4 Notational conventions

The following typographical elements are used in this manual:

`INPUT`	Inputs in examples are shown in bold typewriter font
`Output`	Outputs in examples are shown in typewriter font
	For notes on particularly important information

The metasyntax of ARCHIVE statements is described in the section “Metasyntax” on page 105.

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

References within this manual state the relevant page numbers, as well as the paragraph or chapter where necessary. References to topics described in other manuals contain only the short title of the manual. You can find the relevant place in the text using the index of the appropriate manual.

2 Introduction to ARCHIVE

This chapter introduces you to data saving in BS2000/OSD and to the software product ARCHIVE.

2.1 Data saving in BS2000/OSD

Any operating system must ensure that all stored data is always available. Data saving plays a significant role in ensuring the availability of data: if, for example, data is lost due to an operating error at the processing level a copy of this data should be available, and this copy should be as recent as possible.

The operating system BS2000/OSD supports various methods of producing copies of the current data and thus of saving data.

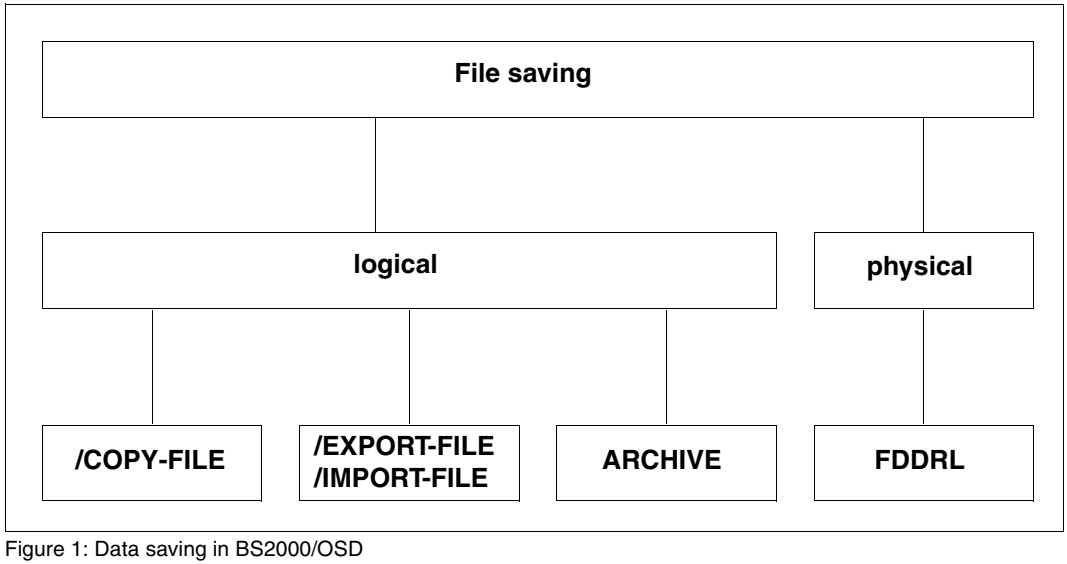


Figure 1: Data saving in BS2000/OSD

There are two types of data saving, namely logical and physical data saving.

In **logical** data saving, files (and job variables) are read from one or more volumes and written as one unit to one or more volumes. Logical data saving thus supports file-specific handling.

In BS2000/OSD, logical data saving is supported by the software products **ARCHIVE** and **HSMS** (see the “HSMS” manual [9]).

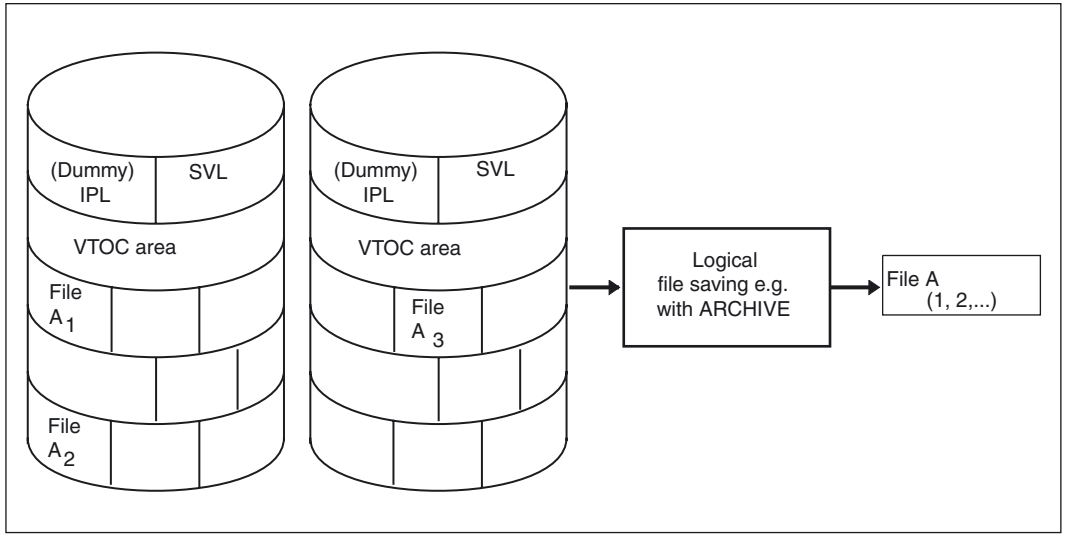


Figure 2: Logical data saving

In **physical** data saving, entire volumes, rather than individual files, are saved: all the data, including the volume header labels of a volume, is written block by block to a second volume, with the result that it has the same structure and contents as the original one.

In BS2000/OSD, physical data saving is supported by the software product **FDDRL** (see the “FDDRL” manual [8]).

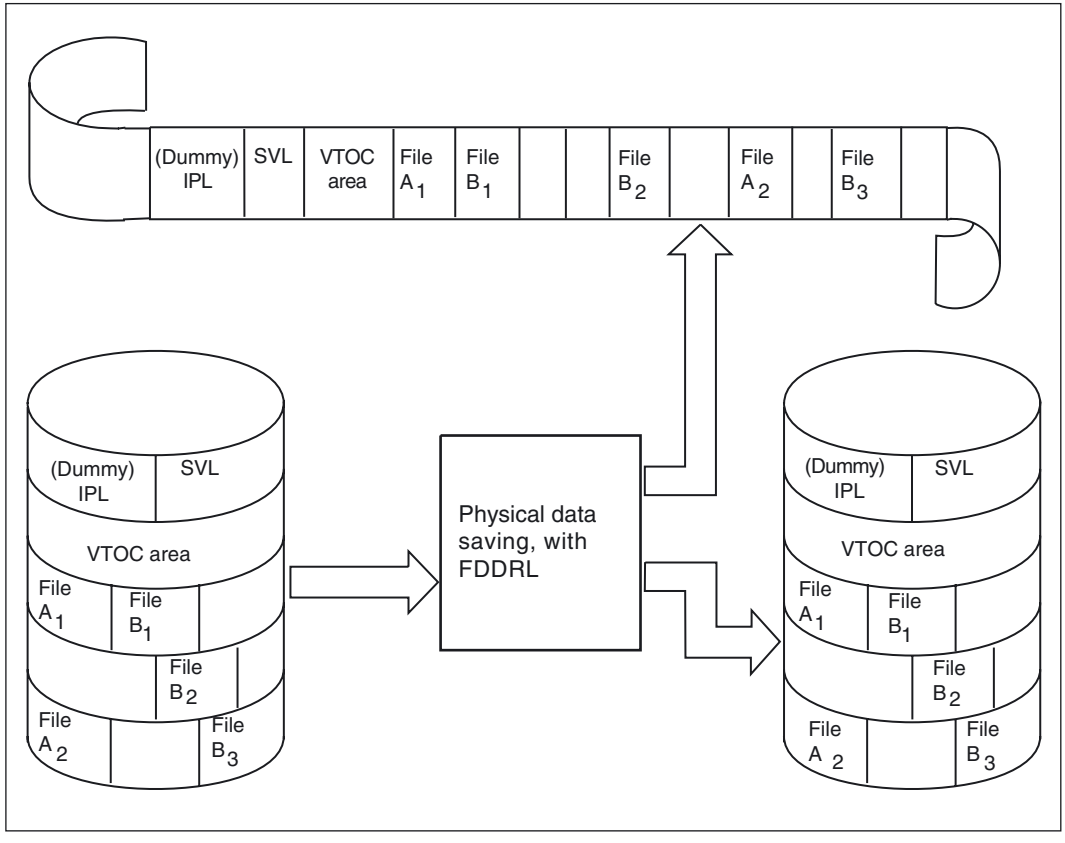


Figure 3: Physical data saving

2.2 Data management with ARCHIVE

The software product ARCHIVE can be used for the following applications:

- **Backup**
  Periodical saving by the system support of all the data on the home pubset and on other public volumes, so that any data lost (disk error or accidental deletion) can be restored in its most up-to-date state (SAVE).
- **Archiving**
  Swapping out from disk to tape (or MT cartridge) of data which is not or no longer needed online. This can be done by
  - a nonprivileged user, for documentation or other purposes;
  - the system support, to relieve the disk storage units of data which has not been used for some time (SAVE statement).
- **Transfer**
  Transfer of data to other BS2000/OSD systems or other user IDs (EXPORT and IMPORT statements).
- **Reorganization**
  The system support can reorganize files by reading saved files back in; these files are then no longer split up over a large number of extents
  (`SAVE CHANGED=NO` and `RESTORE REPLACE=ALLP,SPACE=REORG`).

To support these applications ARCHIVE offers a range of functions, such as:

- **Selection of files and job variables**
  The range of files and job variables to be saved or restored can be specified easily, e.g. by entering partially qualified names or specifying the name of the volume which contains the data.
- **Management via the directory file**
  A directory file is an automatically maintained management directory of the saved data and the associated volumes (for further details, see section “Directory file” on page 37).
- **Incremental save**
  The user can specify that ARCHIVE is to save only the files which have been updated since the last save operation
  (for further details, see section “Full, incremental and partial save” on page 61f).
  Information about the saved files is kept in the directory file.
  An incremental save reduces the execution time and saves storage space.

- **Partial save**
  In the case of large files, the user can specify in a partial save that only the pages which have been updated since the last full save are to be saved
  (for further details, see section “Full, incremental and partial save” on page 61).
  Partial saving is also possible for PLAM libraries, but not for PAM files without PAM keys.
- **Online saving of UDS databases**
  UDS databases can also be saved while they are open
  (for further details, see section “Online saving of UDS databases” on page 85).
- **Support for various save volumes**
  ARCHIVE can save to magnetic tape, MT cartridge or disk. It achieves high data transfer rates because it executes I/O operations with the aid of page chaining, supports streaming mode, and can work with the highest possible blocking factor.
  Since ARCHIVE extends tapes (CONTINUE operand), full use is made of the capacity of these save volumes.
- **Restart option in the event of interruption**
  ARCHIVE optionally writes so-called checkpoints into a file known as the checkpoint file. With the aid of the checkpoints in this file, interrupted ARCHIVE runs can be resumed (for further details, see section “Restarting ARCHIVE processes” on page 82).
- **Parallel processing**
  Saving can be further accelerated by distributing the work over several subtasks running in parallel: when several files are being saved, these are written to different volumes simultaneously
  (for further details, see section “Parallel and serial processing” on page 75).

The software product ARCHIVE can be used for saving and restoring both by systems support (= users with TSOS privilege) and by nonprivileged users.

In interactive and batch modes, the ARCHIVE functions are available once the ARCHIVE subsystem has been loaded (see section “Loading and unloading ARCHIVE” on page 26).

From user programs, the functions can be called by means of the ARCHIVE macro (see chapter “ARCHIVE macro” on page 257).

3 Installation

The software product ARCHIVE must be installed by the system support using the installation monitor IMON. This chapter describes the prerequisites and actions which are necessary for installation.

3.1 System environment and scope of delivery

ARCHIVE V9.0 can be used in the BS2000/OSD operating system V6.0 and higher (for SQ and SX servers in the software package OSD/XC V2.0 and higher) on the current servers from Fujitsu in all hardware series and operates all backup media which have been released for BS2000/OSD V6.0 and higher.

ARCHIVE V9.0 is part of the software product HSMS V9.0.

ARCHIVE V6.0 need be installed with the installation monitor IMON
(see “IMON“ manual [10]).

The following files belong to the ARCHIVE V9.0 scope of delivery (<ver>=090):

- SYSFGM.ARCHIVE.<ver>.D
  Release notice in German.
- SYSFGM.ARCHIVE.<ver>.E
  Release notice in English.
- SYSLIB.ARCHIVE.<ver>
  Macro library containing the ARCHIVE macro
  (for details see the chapter “ARCHIVE macro” on page 257).
- SYSLNK.ARCHIVE.<ver>
  Load module library for the nonprivileged part of ARCHIVE.
- SYSLNK.ARCHIVE.<ver>.TPR
  Load module library for the privileged part of ARCHIVE for use on S servers.
- SKMLNK.ARCHIVE.<ver>.TPR
  Load module library for the privileged part of ARCHIVE for use on SQ servers.
- SPMLNK.ARCHIVE.<ver>.TPR

Load module library for the privileged part of ARCHIVE for use on SX servers.

- SYSNRF.ARCHIVE.<ver>
  NOREF file for the ARCHIVE subsystem.
- SYSPAR.ARCHIVE.<ver>
  Parameter file (SAM) with preset values for most ARCHIVE operands (for details see the section “Parameter file for ARCHIVE” on page 23).
- SYSMES.ARCHIVE.<ver>
  Complete messages file containing the ARCHIVE messages
  (for details see the section “Message files and help texts” on page 22).
- SYSMSH.ARCHIVE.<ver>
  ISAM file with help texts for the HELP command in German and English.

  If IMON is not in use, this file must be shareable under the TSOS ID.
  If IMON is in use, the help texts must be located in the file that is entered in the software configuration inventory in IMON-GPN with the parameters
  `LOGICAL-ID=SYSMSH,INSTALLATION-UNIT=ARCHIVE` (see “IMON“ manual [10]).
- SYSPRG.ARCHIVE.<ver> (ARCHIVE)
  Load module for compatibility with the command `/START-PROGRAM ARCHIVE`.

  If IMON is not in use, the load module must be cataloged as shareable with the file name “ARCHIVE” under DEFLUID (default user ID).

  If IMON is in use, this load module must be located in the file that is entered in the software configuration inventory in IMON-GPN with the parameters
  `LOGICAL-ID=SYSPRG,INSTALLATION-UNIT=ARCHIVE`.
- SYSPRG.ARCHIVE.<ver>.DIRCONV
  Conversion program for directory files.

  If IMON is not in use, the conversion program must be cataloged as shareable with the name DIRCONV under DEFLUID (default user ID).
  If IMON is in use, the conversion program must be located in the file that is entered in the software configuration inventory in IMON-GPN with the parameters
  `LOGICAL-ID=SYSPRG.DIRCONV,INSTALLATION-UNIT=ARCHIVE`.
- SYSRME.ARCHIVE.<ver>.D
  Readme file in German.
- SYSRME.ARCHIVE.<ver>.E
  Readme file in English.
- SYSRMS.ARCHIVE.<ver>
  RMS package for the ARCHIVE subsystem.
- SYSSII.ARCHIVE.<ver>
  File containing information on the structure and installation of ARCHIVE with IMON.

- SYSSDF.ARCHIVE.<ver>
  Syntax file for statements in SDF format (DIRCONV) and the commands /START-ARCHIVE and /START-DIRCONV.
- SYSSSC.ARCHIVE.<ver>
  File containing the subsystem declarations for DSSM

3.2 Generation of the subsystem catalog

ARCHIVE comprises a privileged (TPR) part and a nonprivileged (TU) part. The nonprivileged part of ARCHIVE is loaded by the dynamic linking loader DBL when the ARCHIVE program is started.

The privileged part of ARCHIVE is loaded and unloaded using DSSM (Dynamic Subsystem Management). The systems support staff must therefore declare the ARCHIVE subsystem in the subsystem catalog. The declarations required for this are located in the SYSSSC.ARCHIVE.<ver> file. The generation of the subsystem catalog with SSCM is described in the “Subsystem Management“ manual [7].

ARCHIVE is loaded in the system address space by default. There are, however, also prepared declarations with which systems support can load ARCHIVE in the privileged user address space. The declarations can then be modified as required; notes on how to do this can be found in the SYSSSC.ARCHIVE.<ver> file.

3.3 Parameter service

Systems support can make the following settings for ARCHIVE using system parameters in the parameter service:

1. The system parameter SHUTARCH enables you to determine whether the system should be shut down when ARCHIVE tasks are still running:
   - SHUTARCH=Y
     Once the shut down process has begun, systems support will be asked whether SHUTDOWN should be completed, although ARCHIVE is still being used.
   - SHUTARCH=N
     The system will be shut down although ARCHIVE tasks are still running.
2. The system parameter BACKUP allows the default value for the backup level of files in system backups to be defined.
3. The system parameter FARMTSAV (File Archiving Metadata Save) allows the incremental backup of ARCHIVE to be carried out.

   The default value `FARMTSAV=0` ensures that only files where the data section has been changed are designated for saving. This changes the version number of the file, i.e. the file is saved during incremental backup.

   The default value `FARMTSAV=1` designates all files for saving where the metadata (catalog entry) has changed. The command `/MODIFY-FILE-ATTRIBUTES` also changes the version number of the file in the catalog.
   Thus, the file will be saved during incremental backup even if only the catalog entry has been changed.

3.4 Message files and help texts

The message file `SYSMES.ARCHIVE.<ver>` can be entered in the MIP parameter file on installation with IMON and can be merged into the global message file for unbundled products, which is defined in the system parameter MSGFIL02 (e.g. `SYSMES.EKP.01`).

You will find further information on the ARCHIVE messages in
section “ARCHIVE messages” on page 29.

3.5 Parameter file for ARCHIVE

The parameter file SYSPAR.ARCHIVE.<ver>, which is supplied together with ARCHIVE, contains presettings for the most important ARCHIVE operands. Some operands are omitted either for security reasons (e.g. PASSWORD) or because they are used only infrequently (e.g. CONVERSION for RESTORE).

The SHOW-DEFAULT statement (see page 193) of ARCHIVE contains a list of the ARCHIVE parameters and a description of individual parameters.

The parameter file is assigned in the ARCHIVE subsystem declarations for DSSM as a subsystem information file.

ARCHIVE parameters are only used in ARCHIVE statements if the operand corresponding to the parameter is not specified in the statement (see “Underlining” on page 106).
In the syntax representation, the presetting for the parameters is specified as the default value (see page 194). The value for the parameter can differ from the presetting if the parameter file is altered, see below.

3.5.1 Loading the parameter file

The parameter file is used when ARCHIVE is loaded with DSSM.

Any error in the parameter file will also cause the load operation to be aborted.

3.5.2 Changing the presettings

The system support can change the parameter file presettings with a file editor. Note, however, that the changes become effective only when ARCHIVE is loaded again with DSSM.

When changing the presettings the system support must take great care, because syntax errors in the parameter file will cause the ARCHIVE load operation to be aborted or will cause internal errors during execution.

The following rules should be observed when changing the default settings:

- The parameter file is not checked for completeness. If parameters are missing, the default values will be used.
- An asterisk “*” in column 1 indicates a comment line.
- Only one record per operand is permitted.

- Nothing to the left of the equals sign '=' may be changed.
- The value for a parameter must be entered immediately after the equals sign '='.
- If several definitions are entered for one operand, the last assignment is used.

For safety reasons the system support should log the presettings, either by adding comments to the standard records or by copying the parameter file.

The current operand values can be displayed via the SHOW-DEFAULT statement of ARCHIVE once the subsystem has been loaded.

3.6 Starting a new version of ARCHIVE for the first time

When a new ARCHIVE version is used for the first time, the system support staff must erase the queue file `$TSOS.ARCHIVE.LATER` and the checkpoint file `$TSOS.ARCHIVE.CHKPT` (see section “Work files” on page 45). This means, however, that all queuing or interrupted jobs from the old version are lost.

An ARCHIVE process should then be started with `NOW=NO`. ARCHIVE thus creates a new queue file and a new checkpoint file automatically.

If a number of tasks call ARCHIVE simultaneously during the first start, conflicts can occur during creation of the checkpoint file which lead to a program crash.

If systems support load an ARCHIVE version that does not match the operating system, ARCHIVE will close with one of the following messages: `ARC0007`, `ARC0009` or `ARC0294`. In order to be able to load the current version of ARCHIVE, systems support must first unload the subsystem (see section “Loading and unloading ARCHIVE” on page 26).

4 Execution and environment

This chapter describes

- how ARCHIVE is loaded and unloaded;
- how ARCHIVE is called and the ARCHIVE operating modes;
- the ARCHIVE mssages
- the ARCHIVE processes;
- the procedure control;
- the directory file;
- the work files;
- the outputs from ARCHIVE;
- the interaction between ARCHIVE and MAREN and between ARCHIVE and HSMS.

4.1 Loading and unloading ARCHIVE

The privileged part of ARCHIVE can be loaded in two different ways:

- Systems support (SUBSYSTEM-MANAGEMENT privilege) or the operator can explicitly load the subsystem ARCHIVE with `/START-SUBSYSTEM ARCHIVE`
- All users can load ARCHIVE
  - by calling the TU program with `/START-ARCHIVE`
  - with the command `/START-EXECUTABLE-PROGRAM FROM-FILE=$ARCHIVE` (also, for reasons of compatibility: `/START-PROGRAM $ARCHIVE`)
  - or by calling the ARCHIVE macro (see chapter “ARCHIVE macro” on page 257).

  The TU program is a load module which calls up the privileged part of ARCHIVE via SVC and loads it dynamically if required. However, only the first executable ARCHIVE statement (e.g. STATUS, EXPORT, SAVE etc., but not END, PARAM, FILES and HELP) loads the privileged part of ARCHIVE. In this way, the subsystem is only loaded when it is actually required by the user.

During the load process, object corrections are incorporated from the file `SYSREP.ARCHIVE.<ver>`.

ARCHIVE is unloaded by systems support (SUBSYSTEM-MANAGEMENT privilege) or the operator with `/STOP-SUBSYSTEM ARCHIVE`

ARCHIVE is not unloaded until all ARCHIVE jobs started before the subsystem was deleted have terminated. ARCHIVE can then be loaded again in the same BS2000 session. The ARCHIVE version can therefore be changed during the current session, or new object corrections can be incorporated.

4.2 Calling ARCHIVE

Every user can call ARCHIVE with `/START-ARCHIVE` or
`/START-EXECUTABLE-PROGRAM FROM-FILE=$ARCHIVE`.

Whenever ARCHIVE expects input of a statement during program execution, the user can use the HELP statement (see page 135) to request information about the available statements and about the syntax of the individual statements.

A typical ARCHIVE call could look like this:

1. You call ARCHIVE with `/START-ARCHIVE`.
2. If you do not want to use the default values, you can now use the PARAM statement to select parameter values for this run, e.g. `PARAM OPERATOR=YES`.
3. You then enter FILES and/or JOBVAR statements to specify the files and/or job variables that are to be processed. You may enter several such statements.
4. You then specify in a single statement how these files and job variables are to be processed. You can SAVE, LIST or RESTORE the files and job variables.
5. Steps 2 to 4 can be repeated as often as required before ARCHIVE is terminated.
6. With the K2 key, you can only interrupt the statements INQUIRE and SHOW-DEFAULT.
7. On completion of processing, you terminate ARCHIVE with END.

ARCHIVE parameters are only used in ARCHIVE statements if the operand corresponding to the parameter is not specified in the statement. (see “Underlining” on page 106).
In the syntax representation, the presetting for the parameters is specified as the default value (see page 194). The value for the parameter can differ from the presetting if the parameter file is altered, see below.

You can also call ARCHIVE with the ARCHIVE macro
(see chapter “ARCHIVE macro” on page 257).

4.3 ARCHIVE operating modes

Both systems support (= users with the TSOS privilege) and nonprivileged users can use ARCHIVE. Some functions and operand values are only available to systems support.

ARCHIVE expects all the statements from the system file SYSDTA, which in interactive mode is assigned to the terminal, and in batch mode to the ENTER file.

If ARCHIVE is to be called in a procedure, the system file must be assigned with:

```
ASSIGN-SYSDTA TO-FILE=*SYSCMD
```

If the statements for ARCHIVE are to be read from a SAM or ISAM file, this file must be assigned with:

```
ASSIGN-SYSDTA TO-FILE=filename
```

By default, the following are output on **SYSOUT**:

- error messages and warnings
- messages relating to tapes and MT cartridges for the POOL statement
- information for the STATUS statement about ARCHIVE processes stored in the queue file `ARCHIVE.LATER`.

By default, the following information is output on **SYSLST**:

- the statements entered, together with any resulting error messages or warnings
- the report on the ARCHIVE run
- the end message `ARC0009 ARCHIVE TERMINATED`

Output on SYSLST can be controlled with the LIST operand in the appropriate statements. Additionally or alternatively, it can be redirected to SYSOUT or totally suppressed.

4.4 ARCHIVE messages

The “Meaning” and “Response” texts for a message can be requested by means of the BS2000 command `/HELP-MSG-INFORMATION ARCnnnn`.

You will also find the messages using an HTML application on our manual server (URL: *http://manuals.ts.fujitsu.com*) and on the DVD “BS2000/OSD SoftBooks”.

Using the `MSGMAKER` utility routine, you can list all the messages in the ARCHIVE messages file together with the definition and action texts. You can determine the name of your current ARCHIVE messages file with:

```
/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=ARCHIVE(VERSION=V09.0A),
                        LOGICAL-ID=SYSMES
```

When the ARCHIVE program is called, ARCHIVE issues the following message via SYSOUT:

```
ARC0001 ARCHIVE READY
```

If the syntax check of a statement has been successful, ARCHIVE outputs the following message via SYSOUT and, if desired, additionally via SYSLST:

```
ARC0002 STATEMENT ACCEPTED.
        ARCHIVE SEQUENCE NUMBER 'x.yymmdd.hhmmss', VERSION='9.0A'
```

Once a statement has been completely executed, one of the following messages appears (depending on the execution result):

```
ARC0003 ARCHIVE STATEMENT COMPLETED

ARC0004 ARCHIVE STATEMENT COMPLETED WITH WARNINGS

ARC0005 ARCHIVE STATEMENT COMPLETED WITH ERRORS

ARC0006 ARCHIVE STATEMENT NOT COMPLETED

ARC0007 ARCHIVE STATEMENT REJECTED, VERSION '(&00)'
```

After all statements, including the END statement, have been completely executed, the following message is issued:

```
ARC0009 ARCHIVE TERMINATED
```

ARCHIVE issues the messages with the following numbers at the console:
`ARC0000`, `ARC0018`, `ARC0300`, `ARC07xx` and `ARC09xx`

In addition, messages with the code `ARC07xx` are also issued via SYSOUT.

All other messages are issued to the user, either via SYSOUT or in the ARCHIVE log via SYSLST.

If ARCHIVE issues a message with the code ARC03xx to the user, the operator receives message ARC0300 at the console and has the option of taking a system dump.
After message ARC0900 at the console it is likewise possible to take a system dump.

Regardless of whether file names or alias names are used in statements, the file names are always shown in the messages.

Depending on their contents, ARCHIVE messages may be classified as the following:

- **confirmation messages** (ARC0000 – ARC0019, ARC0033, ARC0047)
  Meaning
  The ARCHIVE run has terminated normally, no job switch is set.
- **warning messages** (ARC0020 – ARC0099, ARC0800 – ARC0839)
  Meaning
  The ARCHIVE statement is executed, job switch 30 is set.
- **minor error messages** (ARC0100 – ARC0799, ARC0900 – ARC0999)
  Meaning
  ARCHIVE has detected an error. The statement is nevertheless executed and job switch 31 is set.
- **serious error messages** (ARC0006, ARC0007, ARC0294)
  Meaning
  ARCHIVE has detected a serious error. The statement is not executed and the spin-off mechanism is activated.

4.5 ARCHIVE processes

ARCHIVE processes input statements in different ways.

- The statements END, PARAM, FILES, JOBVAR and HELP are executed directly by the user task itself in the nonprivileged part.
- For the processing of all other statements, an ARCHIVE process is created and executed in the privileged part.

4.5.1 ARCHIVE sequence number

For statements requiring input/output operations on an ARCHIVE file or on a save volume, an ARCHIVE process is created and executed in the privileged part. ARCHIVE reports this with the following message:

```
%ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'x.yymmdd.hhmmss',
         VERSION='9.0A'
```

The process contains an ARCHIVE sequence number (ASN) in the format `x.yymmdd.hhmmss`, where:

`x` specifies the type of ARCHIVE process. “x” may be any of the following:

A (=active)
If the ARCHIVE run is still in progress or has been interrupted.

L (=later)
The ARCHIVE run was defined with `NOW=NO` and entered in the queue file `ARCHIVE.LATER` (see section “Batch job from the queue file” on page 32).

S (=save)
The ARCHIVE run has been completed. S is entered in the directory file as the save version identifier. identifier (svid).

`yymmdd` date (`yy`=year, `mm`=month, `dd`=day).
ARCHIVE internally converts all two-digit year entries to four-digit entries, which means that there will be no problems at the transition from 1999 to 2000.

`hhmmss` time of day (`hh`=hours, `mm`=minutes, `ss`=seconds).

Nonprivileged users have to specify the ARCHIVE sequence number in the DELETE and PROCESS statements to delete or start ARCHIVE processes of type A or L.
Processes of type S, i.e. save versions, can be deleted using PURGE.

Archive writes the Archive sequence number of the last run to the checkpoint file.
The sequence number comprises the current date and time, so must always be greater than the number for the previous run (stored in the checkpoint file). If it is not possible to use the date and time, the new sequence number comprises
`last sequence number + 1`.

The checkpoint file `$TSOS.ARCHIVE.CHKPT` (or `$TSOS.ARCHIVE.CHKPX` with HSMS operation for UFS files) is stored in the home pubset in SF environment or in the respective SM pubset in SM environment.

4.5.2 ARCHIVE subtask

By default (operand `NOW=YES`), statements involving input/output operations on save volumes (EXPORT, IMPORT, LIST, RESTORE and SAVE) are processed by a task created by ARCHIVE, known as the subtask. In this case, the I/O operations are not performed by the user job (main task).

Example

```
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100814.163022',
           VERSION='9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'OPDJ' GENERATED
```

In interactive mode, the user must wait until the subtask has been completed and the user job has brought the directory file up to date and created the lists before (s)he can make a new entry.

The subtask is started in the same batch category and job class as the main task.

4.5.3 Batch job from the queue file

However, the user has the option of having the statements EXPORT, IMPORT, LIST, RESTORE and SAVE, as well as the statements for managing the directory file (POOL, PURGE), processed by means of a batch process. This is done by specifying `NOW=NO` in the appropriate statement. The batch process then receives an ARCHIVE sequence number with the identifier L.

However, an identifier is assigned only if no queue job has been created under the user job. If several batch processes are created under the same user job, these are stored under a single ARCHIVE sequence number.

The batch process is placed in the queue file (`ARCHIVE.LATER`). You can output it by means of the STATUS statement.

You can start the batch processes in the queue file later with the PROCESS statement or delete them with the DELETE statement.

In order to process the PROCESS statement, an ENTER file is created and a batch job is started. Two or more statements entered in different ARCHIVE calls but under the same job are processed in a single batch job.

In the ENTER file, ARCHIVE is called with `/START-ARCHIVE`.

Example

```
/START-ARCHIVE ------------------------------------------------------------ (1)
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES NA=FILE.1 ---------------------------------------------------------- (2)
*SAVE DIR=ARCHIVE.DIR,TAPES=OP,DEVICE=TAPE-C4
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.103915',
           VERSION='9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN '2JDZ' GENERATED
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '1' PAM-PAGES FOR '1' FILES AND
           '0' JVS IN '0' SECONDS
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*FILES NA=FILE.2 ---------------------------------------------------------- (3)
*SAVE DIR=ARCHIVE.DIR,TAPES=OP,DEVICE=TAPE-C4,NOW=NO
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'L.100810.103931',
           VERSION='9.0A'
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*STATUS ------------------------------------------------------------------ (4)
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.103932',
           VERSION='9.0A'
*ARCHIVE STATUS FOR USER TSOS
   L.100810.103931
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END ---------------------------------------------------------------------- (5)
%  ARC0009 ARCHIVE TERMINATED
/EXIT-JOB


/SET-LOGON-PARAMETERS
/START-PROG $ARCHIVE ------------------------------------------------------ (6)
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
```

```
%  ARC0001 ARCHIVE READY
*FILES NAME=FILE.3 ------------------------------------------------------------ (7)
*SAVE DIR=ARCHIVE.DIR,TAPES=OP,DEVICE=TAPE-C4,NOW=NO
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'L.100810.104845',
           VERSION='9.0A'
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*STATUS ----------------------------------------------------------------------- (8)
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.104846',
           VERSION='9.0A'
*ARCHIVE STATUS FOR USER TSOS
   L.100810.103931
   L.100810.104845
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*PROCESS L.100810.103931 ------------------------------------------------------ (9)
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.104901',
           VERSION='9.0A'
%  JMS0066 JOB '(NONE)' ACCEPTED ON '10-08-10' AT '10:49', TSN = 2JD2
%  ARC0012 ARCHIVE-PROCESS '100810.103931' PROCESSED FROM QUEUE FILE
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*STATUS TYPE=ACTIVE ---------------------------------------------------------- (10)
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.104902',
           VERSION='9.0A'
*ARCHIVE STATUS FOR USER TSOS
   A.100810.104902 2JD1
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*STATUS ---------------------------------------------------------------------- (11)
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.104903',
           VERSION='9.0A'
*ARCHIVE STATUS FOR USER TSOS
   L.100810.104845
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*DELETE L.100810.104845 ------------------------------------------------------- (12)
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.104916',
           VERSION='9.0A'
%  ARC0012 ARCHIVE-PROCESS '100810.104845' DELETED FROM QUEUE FILE
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*STATUS ---------------------------------------------------------------------- (13)
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.104917',
           VERSION='9.0A'
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END ------------------------------------------------------------------------- (14)
%  ARC0009 ARCHIVE TERMINATED
```

(1) The user calls ARCHIVE.

(2) A save run is started with the FILES and SAVE statements and, by default, processed by a subtask. The next entry cannot be made until the run has been processed.
The directory was created previously.

(3) A further save run is created with the specification `NOW=NO` and is written to the queue file `L.100810.103931`.

(4) The STATUS statement shows the ARCHIVE process that has been created.

(5) The user first terminates ARCHIVE, and then terminates the user job with `/EXIT-JOB`.

(6) The user starts ARCHIVE again under a new user job.

(7) The user creates a further save run with `NOW=NO`.

(8) Since the second save run was created under another user job, the STATUS statement now shows two ARCHIVE processes.

(9) The user starts the first save run from the queue file by specifying its sequence number. A batch job is created.

(10) `STATUS TYPE=ACTIVE` shows that a process has been started.

(11) The STATUS statement (with the default value `TYPE=LATER`) shows that the previously started process is no longer in the queue file.

(12) The user deletes the remaining process with the DELETE statement.

(13) The STATUS statement no longer shows any entries.

(14) The user terminates ARCHIVE.

4.6 Information about execution

Information about the execution of the ARCHIVE process can be obtained from the settings of job switches in procedures and ENTER jobs.

ARCHIVE sets these switches during or after the run. The setting of a job switch corresponds to a particular message. If, for instance, job switch 30 has been set, the following message is output:

```
ARC004 ARCHIVE STATEMENT COMPLETED WITH WARNINGS
```

Message number	Message text	Action taken by ARCHIVE
ARC0003	ARCHIVE STATEMENT COMPLETED	The ARCHIVE run has been completed without problems.
ARC0004	ARCHIVE STATEMENT COMPLETED WITH WARNINGS	The ARCHIVE run has been completed with warnings; job switch 30 has been set.
ARC0005	ARCHIVE STATEMENT COMPLETED WITH ERRORS	The ARCHIVE run has been completed with errors; job switch 31 has been set.
ARC0006	ARCHIVE STATEMENT NOT COMPLETED	The ARCHIVE run has been aborted; the spin-off mechanism has been triggered.
ARC0007	ARCHIVE STATEMENT REJECTED, VERSION '(&00)'	
ARC0294	SUBSYSTEM ARCHIVE NOT ENABLED BY DSSM	

If a statement is not executed (ARC0006, ARC0007, ARC0294), the spin-off mechanism is triggered, i.e. once ARCHIVE has terminated, all commands up to the next SET-JOB-STEP or up to the end of the procedure file or ENTER file are skipped.

4.7 Directory file

The directory file is a catalog kept automatically by ARCHIVE of all saved files and job variables as well as the save versions and save volumes on which the data is saved.

The directory file allows ARCHIVE to offer straightforward management of save data and a simple method of restoring lost data. For this reason, you should always execute ARCHIVE save with a directory file. The directory file permits, for example, rapid access to information on saved data without having to access the relevant volumes (INQUIRE). In addition, the following ARCHIVE functions are available only if a directory file is used:

- incremental save and partial save
- requesting the save volumes from a prepared volume pool
- checking the uniqueness of a file name in a save version (see the example on section “Continuation of save versions” on page 81 for a possible exception)
- easy reconstruction of the last save status of a file
- restart after a system crash or device error

The same directory file can be used for save operations (SAVE, EXPORT) to tape, MT cartridge and disk.

If no directory file is specified during a save run, ARCHIVE searches for a directory file with a name that corresponds to the parameter value DIR-S-PO-PU in the parameter file (see page 193).
If this value is not specified in the parameter file, ARCHIVE searches for a directory file with the name `ARCHIVE.DIR`. If no directory file exists with this name, ARCHIVE will issue the message `ARC0157`.

The directory file contains the following administrative information for each saved file or job variable:

- the name of the file or the job variable
- number of the save version identifier (svid);

  this contains the date and time of the save operation
- the VSNs of the save volumes on which the file or job variable is stored

- details of what was saved and the method of saving; ARCHIVE distinguishes between the following save types:

Save type	Meaning
CNS	The file is in the catalog but not saved
CATL	The catalog entry of file was saved
FULL	The entire file was saved
PART	The file was only partially saved
FGGI	The file generation group was saved
MIGF	The file was migrated (see the section “Migrated files”, page 56); the catalog entry was saved
JV	The job variable was saved

- file version identifier from the catalog
- internal file name (CFID)

Managing the directory file

The following ARCHIVE statements are available for managing the directory file:

INQUIRE Output information from the directory file.

PURGE Delete save versions from the directory file.

POOL Add/remove VSNs of magnetic tapes and magnetic tape cartridges to/from the pool of the directory file (see the next section).

4.7.1 Volume pool

The directory file contains a volume pool whose volumes can be used for saving with this directory file. In this pool the user can enter the VSNs of tapes and MT cartridges which are to be used for save operations. The volumes which actually contain save versions are also entered in the pool.

Alternatively, the software product MAREN can be used to designate volumes for use with a specific directory file (see section “ARCHIVE and MAREN” on page 55).

In a save run with a directory file, the volumes required are requested from the pool by default; VSNs need not be specified.
However, the device type must be specified if it is not the standard device type (which is entered in the ARCHIVE parameter file under DEVICE-TAPE-T-C), even if volumes of only one type are entered in the volume pool.

Tapes and MT cartridges are added to the volume pool with the ADD operand of the POOL statement (see page 164). The volumes included in the directory file can be output via the POOL operand of the INQUIRE statement. The report then contains the entry “POOL” in the “OWNER” column (see “example 2” on page 148).

Volumes are also added to the directory file if specific VSNs are specified or are assigned by the operator or via the MAREN free tape allocation facility in a save run with a nonstandard volume assignment. These VSNs can be output by means of the POOL operand of the INQUIRE statement. The report then contains the entry “OPERATOR” in the “OWNER” column.

During saving (SAVE/EXPORT), the RETPD operand can be used to assign a retention period for the save volumes, during which the volumes cannot be overwritten. This retention period is also output when INQUIRE is entered.
At the end of the retention period, the expiration date is reached.

In certain cases the expiration date for a volume which is entered in the ARCHIVE directory or, if it exists, in the MAREN catalog can be higher than the expiration date entered on the volume. This is always the case when, during an incremental save, a file is not saved because it had already been saved in a preceding full save and the expiration date of the dincremental save is higher than that of the full save. The expiration date of the full save/of the volume is then automatically increased in the ARCHIVE directory (and possibly also in the MAREN catalog). See also the section “Retention period and expiration date” on page 63.

Volumes which do not contain a save version can be removed from the pool with the REMOVE operand of the POOL statement (see page 164).
Volumes which do not come from the volume pool are erased from the directory file by ARCHIVE after you delete the save version contained by the volume with the PURGE command.

The volumes of the pool can have various statuses:

- **AVAILABLE**
  The volume is available for save operations (either it has not yet been used or the save file has been deleted and the retention period has expired).
- **IN-USE**
  The volume is occupied by a save file whose retention period has not yet expired.
- **OBSOLETE**
  The retention period of the volume has expired but the save file has not yet been deleted.
- **UNUSABLE**
  The volume cannot be written to due to an error (either it could not be opened during a save run because of invalid labels or an unrecoverable error has occurred during writing).

  If you specify the POOL operand in the INQUIRE statement, these volumes are marked as 'UNUSABLE' in the REMARK column. ARCHIVE will not access these volumes even if they are specified in another save run.
  These volumes must be removed from the pool with the POOL statement before you can add error-free volumes with the same VSNs.

4.7.2 Directory file under TSOS

System backup by means of ARCHIVE is best executed with a directory file and under the ID TSOS (respectively with the privilege TSOS).

Unless otherwise specified, ARCHIVE uses the directory file $TSOS.ARCHIVE.DIR, but a different name can be specified (see section “Creating a directory file” on page 42).
Only the system support can execute save runs with this directory.

Directory files under TSOS differ from directories under other user IDs in the following characteristics:

- Files belonging to a different user ID can be reconstructed only from a directory file that is under TSOS.

  Exception:
  Co-ownership of files of a foreign user ID.

- Even if the directory file is protected by a read password, nonprivileged users can still execute read functions for data held under their own user IDs, i.e. RESTORE and INQUIRE can be used without having to specify the read password for the directory file. Write functions, such as SAVE, are not possible.

The system support should protect the directory file by means of a read password. If (s)he also wants to prevent read access to the directory file with RESTORE and INQUIRE, the directory file must be created under a user ID other than TSOS; but then system backup is not possible.

4.7.3 Creating a directory file

Any user can define a directory file with any desired name in which ARCHIVE enters information concerning the save versions and with the aid of which the user can restore his/her files and job variables.

A directory file can be created with the EXPORT, SAVE and POOL statements. Its name is defined by way of the operand DIRECTORY=filename. In addition, ',NEW' must be specified for the DIRECTORY operand. If ',NEW' is specified for a directory that is not empty, ARCHIVE aborts the run with a corresponding message.

Example

```
/START-ARCHIVE ──────────────────────────────────────────────────────────  (1)
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*POOL DIRECTORY=DIR.TEST,NEW,ADD=OS0310,DEVICE=TAPE-C4 ──────────────────  (2)
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.133604',
           VERSION='9.0A'
%  MARM121 MAREN CATALOG ENTRY 'OS0310'/'0001' MODIFIED
%  ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'OS0310' ADDED TO THE POOL   (3)
%  ARC0003 ARCHIVE STATEMENT COMPLETED
. . .
```

(1) The user calls ARCHIVE.

(2) The POOL statement is used to set up the DIR.TEST directory file. The MT cartridge is added to the volume pool of the directory file.

(3) ARCHIVE reports that the MT cartridge has been added to the volume pool.

It is also possible to create a directory file with /CREATE-FILE or /ADD-FILE-LINK (in which case ARCHIVE uses the link name DIRLINK for the directory).

If you create a directory file with /CREATE-FILE or /ADD-FILE-LINK, you can specify certain file attributes for the directory file; otherwise, only the default values are assigned.
You cannot change the values for

- the access method (ACCESS-METHOD=*ISAM),
- the record format (RECORD-FORMAT=*VARIABLE),
- the key length and position (KEY-LENGTH and KEY-POSITION).

Write access must be permitted for the directory file (ACCESS=WRITE) and the directory file must be empty. In addition, DIRECTORY=NEW must be specified in the next ARCHIVE run.

The directory file is created with or without a PAM key, depending on the default value defined for the system. Even in systems with PAM keys, the directory can be created as a non-key (NK) file, i.e. without PAM keys (see the „Introductory Guide to DMS“ manual [2]).

A new directory file is created with the characteristic `BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK` (DATA-2K or DATA-4K, depending on whether the disk is an NK2 or an NK4 disk). This requires one of the following three conditions to be met:

- system parameter `BLKCTRL=NONKEY` if the disk has PAM keys
- system parameter `BLKCTRL=KEY` if the disk does not have PAM keys
- or the following command must be explicitly specified:
  `/ADD-FILE-LINK LINK-NAME=DIRLINK,FILE-NAME=<directory-name>,`
  `BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK`

Moreover, the user must specify the operand `DIRECTORY=directory-name,NEW` in the subsequent ARCHIVE run.

In the explicit specification, all three operands are mandatory. The user may supplement the `/ADD-FILE-LINK` command by the POOL-LINK and WRITE-IMMEDIATE operands to support NK-ISAM handling.

If the user wants to link an existing directory file with the attribute `BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK` to a user-own ISAM pool, (s)he must enter the `/ADD-FILE-LINK` command together with the additional operand POOL-LINK. If no `/ADD-FILE-LINK` command is entered or the POOL-LINK operand is omitted, the directory file is linked to the task-local standard pool.

Every directory file contains an identifier indicating the mode in which it was created. It can only be used in this mode.However, a directory file created with `CATID=NO` can be converted with the aid of DIRCONV (see the „HSMS“ manual [9]). After conversion, this directory can be used only for runs with `CATID=YES`.

Directory files cannot be used as file generation group files.

It is not possible to perform more than one save run at a time using the same directory file.

You can set up PFA attributes for the directory file (see section “Performant File Access” on page 49). To do this, enter the value `*HIGH`, for example, with the `PERFORMANCE` operand of `/CREATE-FILE` or `/MODIFY-FILE-ATTRIBUTES`. Note, however, that the operand `DISK-WRITE=*IMMEDIATE` must be retained.

Example

```
/MODIFY-FILE-ATTRIBUTES FILE-NAME=<directory-name>,
   SUPPORT=*PUBLIC-DISK(STORAGE-CLASS=*NONE(
   IO-ATTRIBUTES=*PARAMETERS(PERFORMANCE=*HIGH,USAGE=*READ),
   DISK-WRITE=*IMMEDIATE))
```

4.7.4 Shared update conflict when opening a directory file

If a shared update conflict (error DMS0D99) occurs when opening a directory file, the behavior of ARCHIVE depends on whether the call comes from an interactive task or a batch task:

- interactive task
  ARCHIVE attempts to open the directory file again during the maximum wait time. The maximum wait time is defined in the SHUPDW parameter in the parameter file. If the directory file is still locked after the maximum wait time, the ARCHIVE run is aborted with the message ARC0149.
- batch task (e.g. an HSMS server task)
  ARCHIVE attempts to open the directory file again during the maximum wait time. If the directory file is still locked after the maximum wait time, the message ARC0852 is issued at the operator terminal. The run is then terminated or continued, depending on the reponse.

  If the run is continued and the directory file is still locked, ARCHIVE attempts to open the directory file during the maximum wait time, before the message ARC0852 is issued again.

4.7.5 Directory files created with DSAVE or DEXPORT

Directory files created with the DSAVE or DEXPORT statements, that are no longer supported, can no longer be used in backups. They can only be used for restore runs.

Files that were saved using a directory file with DSAVE or exported using a directory file with DEXPORT can be restored with the RESTORE statement. However, the device type of the hard disk where the save file is located must also be specified for the DEVICE operand in the RESTORE statement.

4.8 Work files

ARCHIVE work files can be divided into two categories:

- files under the ID TSOS which exist only once in the system and are created in the first ARCHIVE run (checkpoint and queue files)
- files which are created under the user ID of the caller in every ARCHIVE run (e.g. the save files).

The files are created in the default pubset of the relevant user ID either with or without a PAM key, depending on the system presetting (system parameter BLKCTRL).

The values for BUFFER-LENGTH and BLOCK-CONTROL-INFO also depend on the disk type, i.e. they are either (STD,1) for K and NK2 disks or (STD,2) for NK4 disks.

In the case of a RESTORE/IMPORT run, the disk save file must also be in the default pubset.

4.8.1 Checkpoint file

The checkpoint file **$TSOS.ARCHIVE.CHKPT** is created with this file name during the first ARCHIVE run. It must remain under this user ID and its name may not be changed. The checkpoint file contains an entry for each current save version and, if the run was carried out with RESTART=YES, restart points, i.e. checkpoints written by ARCHIVE from which abnormally terminated ARCHIVE runs can be restarted. Information is also written to this file (even with RESTART=NO) in the case of RESTORE runs for partially saved files. The file ARCHIVE.CHKPT is cataloged under TSOS. It can be deleted at any time by the system support if it is too large and contains no tasks (ARCHIVE or HSMS) which have to be restarted.

The checkpoint file is created by ARCHIVE with a read password. The system support can change this password with /MODIFY-FILE-ATTRIBUTES. If the file does not have this read password, ARCHIVE will not accept it as a valid checkpoint file.

If the checkpoint file has become too large and does not contain any processes waiting to be started (ARCHIVE or HSMS), the system support can delete it (DELETE-FILE ...,IGNORE-PROTECTION=*READ-PASSWORD).

After specification of the read password, the checkpoint file can be reorganized using PERCON (see the “PERCON” manual [13) even if it is not empty. Prior to this, however, the system support must replace the read password assigned by ARCHIVE by another one (see the first note above in this list).

You can set up PFA attributes for the checkpoint file (see section “Performant File Access” on page 49). To do this, enter the operand PERFORMANCE=*HIGH in /MODIFY-FILE-ATTRIBUTES command. Note, however, that the operand DISK-WRITE=*IMMEDIATE must be retained.

Any alias name is ignored if the complete name of the checkpoint file is specified.

4.8.2 Queue file

The queue file **$TSOS.ARCHIVE.LATER** is created during the first ARCHIVE run for which the operand NOW=NO is specified. This file contains all the ARCHIVE processes defined with NOW=NO, which can subsequently be started from this file with the PROCESS statement or deleted with the DELETE statement.

The notes above for checkpoint files apply analogously to the queue file.

4.8.3 ENTER file

An ENTER file with the name **ARCHIVE.L.date.time** is created for a LATER process when the latter is started with the PROCESS statement.
Sufficient public space must be available for this file. After normal completion of the process, ARCHIVE deletes this file. In the event of a system crash, the system support must delete this file, since it is protected by ARCHIVE with a password.

If temporary files are permitted in the system, the ENTER file is, however, created as a temporary file and deleted automatically.

4.8.4 Print file

A print file with the name **ARCHIVE.P.date.time** is created for every ARCHIVE run with the restart option (`PARAM RESTART=YES`) and `LIST≠NONE`. This print file is created under the user ID under which the run was started and contains information about the run together with an alphabetical list of the files processed in this run. Sufficient public space must be available for this file.

This file is required by ARCHIVE for restarting via the PROCESS statement. It is automatically deleted by ARCHIVE in the case of normal termination or after a DELETE statement for the corresponding ARCHIVE process. If neither a PROCESS nor a DELETE statement is issued for the process, the file is not deleted.

In the case of `RESTART=NO`, the print file is created as a temporary file if temporary files are permitted by the system. Otherwise it is created in the same way as for `RESTART=YES`.

For the print file, the user can PFA attributes (see section “Performant File Access” on page 49) with the link name `PRNTLINK` and, for example, the following command:

```
/ADD-FILE-LINK FILE-NAME=*DUMMY,LINK-NAME=PRNTLINK,
     SUPPORT=*DISK(IO-ATTRIBUTES=*PARAMETERS(PERFORMANCE=*HIGH,USAGE=*READ))
```

4.8.5 Save files

ARCHIVE.SAVE.FILE(date-time-subsave#-O)
Name given to the save file in save runs to tape or MT cartridge. As well as the date and time of the save, the file name also contains the subsave number (3-digit) of the run in which it was created (see also “Parallel processing” on page 76).
'O' indicates that it is an output file.

After normal termination of ARCHIVE, the catalog entry is automatically deleted. Otherwise, the user must delete the catalog entry for this file.

ARCHIVE.SAVE.FILE(date-time-subsave#-I)
is the name given to the save file in restoration runs from tape or MT cartridge. As well as the date and time of the save, the file name also contains the subsave number of the run (3-digit) in which it is processed.
'I' indicates that the file is an input file.

After normal termination of ARCHIVE, the catalog entry is automatically deleted. Otherwise, the user must delete the catalog entry for this file.

ARCHIVE.SAVE.FILE.date.time.vsn
Name given to the save file in save runs to (private or public) disk. It contains the saved files and job variables.

In an EXPORT run, the catalog entry of the save file is deleted from the system catalog if no directory file was specified and the statement was executed successfully.
In an IMPORT or LIST run, the save file is automatically imported by ARCHIVE.
After a SAVE run, the catalog entry is not deleted.

Compatibility of save files

Save files that are created with ARCHIVE V9.0 can be read by all ARCHIVE versions as of V2.8.

Save Files on disk will be created in a neutral NK file format as of ARCHIVE V7.0, independently of the format of the pubset or the private disk. Therefore, continuation of disk save files from earlier versions is not possible.

4.8.6 Performant File Access

The concept Performant File Access (PFA) / HIPERFILE of BS2000/OSD offers the user a number of possibilities for enhancing performance with respect to file access and file processing (see „Introductory Guide to Systems Support“ [3]).

In ARCHIVE, PFA attributes can be specified for directory, checkpoint and print files, i.e. files that are frequently accessed. ARCHIVE retains the PERFORMANCE attributes, if any.

PFA attributes can be defined for:

- checkpoint files and existing directory files, by modifying the catalog:
```
/MODIFY-FILE-ATTRIBUTES FILE-NAME=<filename>,
      SUPPORT=*PUBLIC-DISK(STORAGE-CLASS=*NONE(
      IO-ATTRIBUTES=*PARAMETERS(PERFORMANCE=*HIGH/*VERY-HIGH)))
```
- print files, by using the link name PRNTLINK, which must be entered in the TFT (task file table):
```
/ADD-FILE-LINK LINK-NAME=PRNTLINK,...,
      SUPPORT=*DISK(IO-ATTRIBUTES=*PARAMETERS(PERFORMANCE=...))
```
- new directory files, by using the link name DIRLINK, which must be entered in the TFT (task file table):
```
/ADD-FILE-LINK LINK-NAME=DIRLINK,...,
      SUPPORT=*DISK(IO-ATTRIBUTES=*PARAMETERS(PERFORMANCE=...))
```

4.9 Output from ARCHIVE

ARCHIVE outputs a command listing and a report for each ARCHIVE run.

4.9.1 Command listing

ARCHIVE outputs the statements entered for each ARCHIVE run in a list of statements (command listing); unless otherwise specified the list is output to SYSLST (or to SYSOUT in the case of INQUIRE). The output destination can be controlled by means of the LIST operand.
ARCHIVE checks the syntax of the input statements; if it is correct, the run is started.

Example of a SAVE run

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES NAME=(FILE.1,FILE.2,FILE.3) ------------------------------------------ (1)
*SAVE DIRECTORY=ARCHIVE.DIR,LIST=SYSOUT,TAPES=OP,DEVICE=TAPE-C4 ------------- (2)
*     S A V E   C O M M A N D   L I S T I N G *** ------------------------- (3)

PARAMETER VALUES:- ---------------------------------------------------------- (4)
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = NO
STREAM  = YES

FILES NAME=(FILE.1,FILE.2,FILE.3) ------------------------------------------- (5)
SAVE DIRECTORY=ARCHIVE.DIR,LIST=SYSOUT,TAPES=OP,DEVICE=TAPE-C4
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.140853'
           VERSION='9.0A' -------------------------------------------------- (6)
%  ARC0033 ARCHIVE SUBTASK TSN '2JH6' GENERATED
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '3' PAM-PAGES FOR '3' FILES AND
           '0' JVS IN '0' SECONDS
```

(1) ARCHIVE is to process the files `FILE.1`, `FILE.2` and `FILE.3`.

(2) ARCHIVE is to save the specified files; the directory file `ARCHIVE.DIR` is to be used for the run. The magnetic tape cartridges to be used for the run should be made available by MAREN or the operator.

(3) Header line of the statement listing for the SAVE run.

(4) The parameter values for the SAVE run are listed. The PARAM default values are valid for this ARCHIVE run (see the PARAM statement, page 160).

(5) ARCHIVE logs the two entered files a second time.

(6) ARCHIVE confirms that the input statements are free of syntax errors. The ARCHIVE run is assigned an ARCHIVE sequence number.

Example of a RESTORE run

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES NAME=(FILE.1,FILE.3) ------------------------------------------------  (1)
*RESTORE DIRECTORY=ARCHIVE.DIR,LIST=SYSOUT,REPLACE=YES ---------------------  (2)
*     R E S T O R E   C O M M A N D   L I S T I N G *** -------------------  (3)

PARAMETER VALUES:- ---------------------------------------------------------  (4)
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
SNR     = YES
CATID   = NO
STREAM  = YES

FILES NAME=(FILE.1,FILE.3) -------------------------------------------------  (5)
RESTORE DIRECTORY=ARCHIVE.DIR,LIST=SYSOUT,REPLACE=YES
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100810.141034'
           VERSION='9.0A'--------------------------------------------------  (6)
%  ARC0033 ARCHIVE SUBTASK TSN '2JH7' GENERATED
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '2' PAM-PAGES FOR '2' FILES AND
           '0' JVS IN '0' SECONDS
```

(1) ARCHIVE is to process the files `FILE.1` and `FILE.3`.

(2) ARCHIVE is to restore the specified files; the directory file `ARCHIVE.DIR` is to be used for the run; any existing files with the same names are to be overwritten by the restored files.

(3) Header line of the statement listing for the RESTORE run.

(4) The parameter values for the RESTORE run are listed. The PARAM default values are valid for this ARCHIVE run (see the PARAM statement, page 160).

(5) ARCHIVE logs the two input statements a second time.

(6) ARCHIVE confirms that the input statements are free of syntax errors. The ARCHIVE run is assigned an ARCHIVE sequence number.

4.9.2 ARCHIVE report

For each ARCHIVE run, ARCHIVE outputs a report; unless otherwise specified, this report is output to SYSLST (or to SYSOUT in the case of INQUIRE). The output destination can be controlled by means of the LIST operand.

The report provides information on the following:

- type of run
- output volumes
- input volumes (if there are more than one, the first one)
- the files and job variables processed
- size of the job variables, last-page pointer of the file
- version number of the file from the catalog entry
- save type
- date of the ARCHIVE run
- save version identifier
- user ID

Example of the report for a SAVE run

```
     ***          S A V E   A R C H I V E   R E P O R T  ***          10-08-10       PAGE 1 --- (1)
                        SAVE VERSION IDENTIFIER - S.100810.140853 - ON VOLUME TYPE   TAPE-C4
  SUBSAVE
   NUMBER       VSNS

     0        OS0424

     ***          S A V E   A R C H I V E   R E P O R T  ***          10-08-10       PAGE 2 --- (2)
                        SAVE VERSION IDENTIFIER - S.100810.140853 - ON VOLUME TYPE   TAPE-C4
                             ***   USER - TSOS       ***
  FILE/JOB VARIABLE NAME                LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                                  VERS     SIZE  TYPE    VSN  TYP SAVE  VSN(S)

FILE.1                                       1        1  FULL  SBZ3.2  D     0  OS0424
FILE.2                                       1        1  FULL  SBZ3.1  D     0  OS0424
FILE.3                                       1        1  FULL  SBZ3.2  D     0  OS0424

     ***       E N D   O F   A R C H I V E   R E P O R T  ***          10-08-10       PAGE 3 --- (3)
```

(1) ARCHIVE report of the SAVE run; the report is displayed on the screen since `LIST=SYSOUT` was specified.

The first page of the ARCHIVE report contains the following information:

– SAVE VERSION IDENTIFIER
Number of the save version (svid), to which the files are saved.

– SUBSAVE NUMBER
  Number of the parallel run
  In the example, only one run was defined; it has the number 0.

– VSNs
  VSNs of the volumes used by ARCHIVE for the run.
  If these files or file blocks occupy more VSNs than are provided for in the ARCHIVE report, three plus signs (+++) are written in place of further VSNs.

(2) The second page of the ARCHIVE report contains the following information:

– USER
  User ID (TSOS).

– FILE/JOB VARIABLE NAME
  Names of the files and job variables on the tape.
  Files which could not be saved due to an error are listed at the beginning of the report.

– VERS
  Version number of the saved file from the catalog entry.

– LASTPG/SIZE
  Last-page pointer for files; size for job variables.

– SAVE VERSION IDENTIFIER
  Number of the save version (svid).

– SAVE TYPE
  Save type: JV, FULL, PART, FGGI, CATL, CNS, MIGF
  (the overview of the save types on page 38 shows the meanings).

– INPUT VSN
  VSN of the first input volume.

– DEV TYP
  Device type of the input volume: D=disk, T=tape, C=catalog.

– SUBSAVE
  Number of the parallel run.

– OUTPUT VSN(S)
  Output volume(s).

(3) Final page of the report on the SAVE run.

Example of the report for a RESTORE run

```
     ***     R E S T O R E   A R C H I V E   R E P O R T  ***         10-08-10       PAGE 1 --- (1)
                          SAVE VERSION IDENTIFIER = S.100810.140853
  SUBSAVE
   NUMBER       VSNS

     0        OS0424

     ***     R E S T O R E   A R C H I V E   R E P O R T  ***         10-08-10       PAGE 2 --- (2)
                           ***   USER = TSOS       ***
  FILE/JOB VARIABLE NAME                 LASTPG/   SAVE VERSION  SAVE  INPUT   SUB OUTPUT
                                   VERS     SIZE    IDENTIFIER   TYPE   VSN   SAVE DISK(S)

FILE.1                                        1        1  100810.140853  FULL  OS0424   0  SBZ3.2
FILE.3                                        1        1  100810.140853  FULL  OS0424   0  SBZ3.2

     ***       E N D   O F   A R C H I V E   R E P O R T  ***         10-08-10       PAGE 3 --- (3)
```

(1) ARCHIVE report of the RESTORE run; the report is output to the terminal as `LIST=SYSOUT` was specified.

The first page of the ARCHIVE report contains the following information:

– SAVE VERSION IDENTIFIER
Number of the save version (svid) from which the files have been restored.

– SUBSAVE NUMBER
Number of the parallel run
In the example, only one run was defined; it has the number 0.

– VSNs
VSNs of the volumes which ARCHIVE uses for the run.

If these files or file blocks occupy more VSNs than are provided for in the ARCHIVE report, three plus signs (`+++`) are written in place of further VSNs.

(2) The second page of the ARCHIVE report contains the following information:

– USER
User ID (`TSOS`).

– FILE/JOB VARIABLE NAME
Names of the files and job variables on the tape.

– VERS
Version number of the saved file from the catalog entry.

– LASTPG/SIZE
Last-page pointer in the case of files; size in the case of job variables.

– SAVE VERSION IDENTIFIER
Number of the save version (`svid`).

- SAVE TYPE
  Save type: JV, FULL, PART, FGGI, CATL, CNS, MIGF)
  (the overview of the save types on page page 38 shows the meanings).
- INPUT VSN
  Volume serial number of the first input volume.
- DEV TYP
  Device type of the input volume (D=disk, T=tape, C=catalog).
- SUBSAVE
  Number of the parallel run.
- OUTPUT DISK(S)
  Output volume(s).

(3) Final page of the report on the RESTORE run.

4.10 ARCHIVE and MAREN

MAREN is a software product for the management of volumes in data centers. The volume pools of ARCHIVE can also be subjected to management by MAREN. In this case, MAREN assigns the volumes to an ARCHIVE directory. The MAREN entry of the volume shows the directory assignment; the volume is not available for access from another source (see the “MAREN” manual [12]).

The ARCHIVE statements SAVE, EXPORT, PURGE and POOL automatically call MAREN via an interface, and MAREN updates its management information accordingly. This does not require use of the directory file. Volume management by ARCHIVE can thus be integrated into the volume management of the whole data center.

In order to avoid inconsistencies between MAREN and ARCHIVE directories, MAREN must always be loaded while ARCHIVE is being used. In addition, the information on ARCHIVE tapes contained in the MAREN catalog must not be modified directly by means of MAREN.

Recommendation

If MAREN has been installed, the volume pool should be managed only with the aid of MAREN and not with ARCHIVE.

Any inconsistencies of status between a MAREN catalog and an ARCHIVE directory file can be rectified with the DIRCONV statement `UPDATE-VOLUME-CATALOG` , see the “HSMS” manual [9].

4.11 ARCHIVE and HSMS

HSMS (Hierarchical Storage Management System) is a software product used for backup operations and to support data management on external storage media under BS2000/OSD.

HSMS uses ARCHIVE to copy files between the different storage levels. All I/O operations initiated by HSMS are executed via ARCHIVE runs, and all entries in the directories are made via ARCHIVE.
HSMS manages the data in save files and archive directories which are largely compatible with those of ARCHIVE, as long as functions not specific to HSMS are used (see the “HSMS” manual [9]).

We strongly recommend that you use either only ARCHIVE or only HSMS. If you use both ARCHIVE and HSMS, incompatibilities may result unless you work with different directory files. For instance, a SAVE or RESTORE run under ARCHIVE is not possible with a directory file which was created with HSMS or which is administered by HSMS.

Migrated files

HSMS supports migrated files. A migrated file is one whose data is written to a save file and deleted at the processing level (in order to make space there), but whose catalog entry is retained. The file is flagged in the catalog as migrated; it continues to be addressable via the catalog, even in the case of access attempts via the BS2000/OSD Data Management System (DMS). Here, too, HSMS and ARCHIVE must be loaded in the system.

ARCHIVE can process a migrated file with SAVE under the user ID TSOS. In this case, only the catalog entry of the file with the save type MIGF is saved.
It is not possible to process migrated files with EXPORT; any attempt to do so will result in error messages.

All users are permitted to use RESTORE and IMPORT on migrated files. Migrated files cannot be renamed with RENAME. The SPACE operand is ignored in the case of migrated files.

5 ARCHIVE functions

The functions of ARCHIVE can be divided up into:

Main functions

- **Backup, archiving and restoration of files and job variables**
  with the statements SAVE and RESTORE for the save volumes “tape”, “MT cartridge” and “disk”.
- **Transfer of files and job variables**
  between different BS2000/OSD installations and user IDs by means of the statements EXPORT and IMPORT for the save volumes “tape”, “MT cartridge” and “disk”.

  Transfer of private volumes for saving and incorporating catalog entries for files with the EXPORT and IMPORT statements (`CATONLY=YES` operand).

Supporting functions

- processing of pubsets
- parallel and serial processing of user jobs
- continuation (extension) of save versions
- restarting of ARCHIVE processes
- PAM key handling
- data compression
- online saving of UDS databases
- processing the user definition file
- management of file attributes
- handling of tapes, MT cartridges and disks
- processing of alias names
- handling of duplicate files and job variables

These functions are described in detail in the various sections of this chapter.

5.1 Backup, archiving and restoration

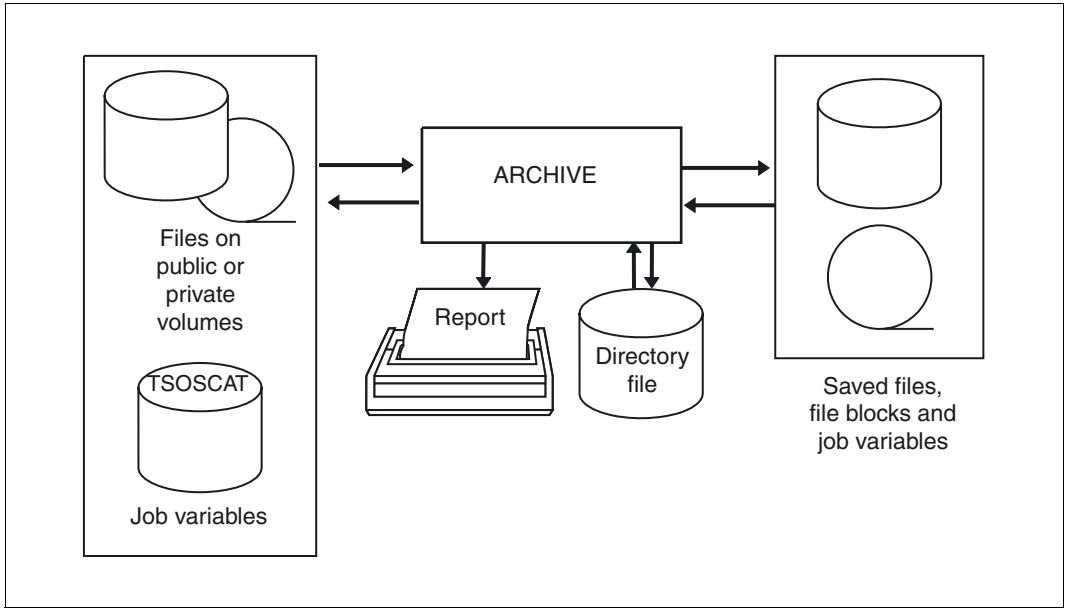


Figure 4: Backup, archiving and restoration of files and job variables

ARCHIVE can be used to save files, file blocks, and catalog entries of files and job variables on tape, MT cartridge or disk. The data is saved in so called save files.

Only the system support can save data to public volumes; saving is only possible to disks of the home pubset.
In the case of files on tape or MT cartridge only the catalog entries are saved.

Files that are saved or restored and save files cannot be reserved by means of the `/SECURE-RESOURCE-ALLOCATION` command.

By default, all the important information concerning a save run is printed out and stored in the directory file. If you wish to read in your files again from the save volume, you need only specify the name of the directory file. All the necessary information, such as the VSN of the save volume, is then extracted from this directory file.

Execution of a save run

`/START-ARCHIVE`	The program is called.
`*FILES ...`	The files to be saved are selected.
`*JOBVAR ...`	The job variables to be saved are selected.
`*SAVE ...`	The directory file name, the VSNs of the save volumes and the operands for the save run are specified.
`*END`	The program is terminated.

Execution of a restoration run

`/START-ARCHIVE`	The program is called.
`*FILES ...`	The files to be read in are selected.
`*JOBVAR ...`	The job variables to be read in are selected.
`*RESTORE ...`	The name of the directory file, the VSNs of the save volumes and the operands for the restoration run are specified.
`*END`	The program is terminated.

ARCHIVE parameters are only used in ARCHIVE statements if the operand corresponding to the parameter is not specified in the statement. (see “Underlining” on page 106).
In the syntax representation, the presetting for the parameters is specified as the default value (see page 194). The value for the parameter can differ from the presetting if the parameter file is altered, see below.

5.1.1 File selection options (FILES statement)

Files can be selected for an ARCHIVE run by specifying

- fully or partially qualified file names (NAME operand)
- ranges of fully or partially qualified file names (NAME, EXCEPT and THRU operands)
- volumes (FROM operand)
- file attributes, which can be specified with the CHANGED, BACKUP and UNUSED operands in the SAVE statement. File attributes comprise information on:

  a) whether or not the file has been modified since the last save.
  Only files modified since the last save run are saved if the `CHANGED=YES` operand is specified.

  b) whether only pages that contain changes are to be saved.
  If `CHANGED=YES,LARGE` is specified, only the pages modified since the last full save operation are saved. Before this can be done, however, the attribute “LARGE” must be entered in the catalog with the command `/MODIFY-FILE-ATTRIBUTES ...,SAVE-PAGES=MODIFIED-PAGES`.

  c) how frequently a file is to be saved (BACKUP operand).

  d) when a file was last accessed to be read or changed (UNUSED operand).

5.1.2 Job variable selection options (JOBVAR statement)

Job variables can be selected for an ARCHIVE run by specifying

- fully or partially qualified job variable names (NAME operand)
- ranges of job variables (NAME, EXCEPT and THRU operands)

Since job variables are a special kind of catalog entry, the CATONLY, CHANGED and BACKUP operands cannot be used when saving job variables. If they are specified despite this, ARCHIVE ignores them and issues a warning message.

5.1.3 Saving files and job variables (SAVE statement)

By means of the SAVE statement, the nonprivileged user or the system support can save files (even encrypted ones), modified data blocks, job variables and catalog entries of files on private volumes (disks, tapes or MT cartridges). The intervals at which saving is to be performed can be defined by the user, e.g. by means of ARCHIVE calls in repeat jobs. ARCHIVE save data can be copied in a SAVE run.

5.1.3.1 Full, incremental and partial save

A **full save** saves all files specified in the FILES statement completely, regardless of whether or not they have been changed since the last save.

An **incremental save** does not save all files indiscriminately. ARCHIVE first checks each file to be saved to determine whether it is new or has been changed since the last save. If ARCHIVE detects that the current contents of a file were already saved during the last save, the file is not saved again. Instead it is registered in the directory as “cataloged but not saved” (CNS).

Incremental saves can result in a considerable reduction of the time and storage space required for system backup carried out by the system support. For this reason, ARCHIVE executes system backup as an incremental save by default.

Full saves and incremental saves can be carried out on different volumes (e.g. full save to magnetic tape and incremental save to private disk).

Up to 255 incremental saves can be performed between any two full saves. After 255 incremental saves, ARCHIVE automatically executes a full save even if otherwise specified in the corresponding statement.

The type of save is controlled via the CHANGED operand of the SAVE statement. Specifying `CHANGED=YES` results in an incremental save, while `CHANGED=NO` causes a full save to be executed.

The scope of an incremental save can be further reduced by a **partial save**. Just as in an incremental save, the files which have remained totally unchanged since the last save are not saved. Those files that have been changed are checked by ARCHIVE to determine the parts (2-K blocks) that have been modified since the last full (not partial) save. Only these are saved in this run.

This means that both the most recent partial save and the most recent full save are required in order to restore a partially saved file.

A partial save partially saves either all files with LARGE in their catalog entry (`CHANGED=YES,LARGE`) or files of a specific size irrespective of the catalog entry (`CHANGED=YES,integer`).

Partial save is not possible for PAM files without a PAM key (BLOCK-CONTROL-INFO=NO), except in the case of PLAM libraries.

Incremental Save and Retention Period Update

Rare cases may occur, when a file is saved full within an incremental save, although the same file version exists in an older full save. In this case, the update of retention period and expiration date for the old full save is suppressed, see the section “Retention period and expiration date” on page 63.

We recommend that you do not use the optional reps any more for suppression of the retention period and expiration date update (ARCHIVE and HSMS operation)

5.1.3.2 File generation groups

ARCHIVE also supports file generation groups (FGGs). These are files which are chronologically connected and have identical file attributes. FGGs are mainly intended for the long-term storage of large data sets, e.g. of data from processing sequences that recur at regular intervals. FGGs thus also afford convenient support for data saving in accordance with the grandfather-father-son principle.

Each file generation is handled as a separate file. Both entire file generation groups and individual file generations can be saved with the usual save facilities (with or without directory, CHANGED=YES, etc.). However, partial saves are not possible.

The following should be noted when saving FG(G)s:

1. A file generation is saved if the following statement is entered:
   FILES NAME=<file-generation-group-name>(*n)
2. A file generation group is saved if the following statement is entered:
   FILES NAME=<file-generation-group-name>

   The generation index and all entries are processed:

   - If CHANGE=NO, the generation index and the entries are saved.
   - If CHANGE=YES, two cases have to be distinguished:
     – If the first and the last entry of the FGG have not been updated since the last full save, the generation index is not saved; the entries are saved FULL or CNS (see section “Directory file” on page 37), depending on whether or not they have been updated since the last full save.
     – If the first or the last entry of the FGG have been updated since the last full save, the generation index and all entries are saved.
   - If the ERASE operand is specified in a save run, the generation index is not deleted if not all entries have been saved.

5.1.3.3 File backup frequency

The BACKUP operand can be used to control how often a file is to be saved. In a save run, all files with a catalog BACKUP entry less than or equal to the specification in the BACKUP operand in the SAVE statement are saved.

Example

The system support uses `BACKUP=A` for day-to-day saving, `BACKUP=B` for weekly saving and `BACKUP=C` for monthly saving. The user must then enter `BACKUP-CLASS=A` in the `/MODIFY-FILE-ATTRIBUTES` command for daily saving, `BACKUP-CLASS=B` for weekly saving, etc.

The default value for the BACKUP operand is `D`. If you want to exclude files from a backup with SAVE, you are recommended to assign these files `BACKUP-CLASS E` and only to use the values `A` through `D` for the BACKUP operand.
An exception here is formed by directory files with the BACKUP entry `E`: When `DIRSAVE=YES` is specified for an EXPORT or SAVE run and a save takes place with `BACKUP=D`, the directory file is saved nevertheless.

5.1.3.4 Save scope

Files are only saved as far as the last written page (LAST PAGE) and not in accordance with the space allocation (FILE SIZE), unless the file has not been closed correctly.

For reasons of performance, ARCHIVE saves only blocks with a valid internal file name (CFID).

5.1.3.5 Save version

Files which have been saved at a certain time are combined to form a save version. A save version is identified by a save version identifier (svid). This is assigned by ARCHIVE in the form `S.yymmdd.hhmmss`, where `yymmdd` is the date (`yy`=year, `mm`=month, `dd`=day) and `hhmmss` the time of day (`hh`=hour, `mm`=minute, `ss`=second).

Save versions are addressed via the svid in ARCHIVE statements.

Please refer also to the section “Performance of copy processing” on page 103.

5.1.3.6 Retention period and expiration date

Data backups should be kept available for a particular period of time.
The retention period is the period for which the save version in the directory file is to be protected against deletion using the PURGE statement. During the retention period the volume may not be overwritten, and consequently the data it contains may not be deleted. The retention period is defined in the SAVE or EXPORT statement using the RETPD operand.

The expiration date is reached when the retention period has elapsed, in other words Expiration date = Creation date + Retention period (in days).

The expiration date is noted on the volume concerned, in the directory file and possibly in the MAREN catalog. In MAREN not only the retention period of the volume is noted, but also the name of the directory file. When the retention period has expired, the save is regarded as “obsolete” and can be deleted using the PURGE statement. When deletion takes place, MAREN is automatically informed and deletes the name of the directory file in the MAREN entry for the volume.

If, during an incremental save (SAVE statement with CHANGED=YES), it is detected that a files has not changed since it was last saved, it is entered in the directory file as Cataloged-Not-Saved (CNS). If the expiration date of the last save which this file contains is older than the expiration date of the current incremental save, the retention period of the last save is updated. The last save is thus assigned the same expiration date as the current incremental save; there is no danger that the file (of the save type FULL) saved earlier will be released before the incremental save (save type CNS).

When the retention period is automatically incremented as described, the expiration date is also incremented in the MAREN entry for the volume.
The retention period cannot be updated on the volume itself. If MAREN is **not** being used, other applications can overwrite the tape.

The retention period is, however, **not** automatically incremented if the last backup of the file is too far in the past (i.e. more than a third of the retention period specified in the current save run, but at least 7 days). In this case the file concerned undergoes a full save (save type FULL) despite the incremental save with CHANGED=YES. This ensures continuous release of save versions and tapes up to the expiration date while guaranteeing reliable restorability.

5.1.3.7 Support for files larger than 32 Gbyte

Hard disks and pubsets can be set up that allow files larger than 32 Gbyte (referred to as “large files”). These “large files” are supported by ARCHIVE.

ARCHIVE supports large files in as compatible a form as possible, without extensions to the user interface. Large files are ignored with failure messages in RESTORE if they cannot be restored. Large files are ignored when duplication of save versions is carried out for BS2000/OSD-BC < V5.0.

Files are saved as “small” or “large” files, according to their actual size, regardless of the size of the hard disk.

The term used here for large files is independent of the LARGE operand in the CHANGE parameter of the SAVE statement. “tape”

5.1.3.8 Miscellaneous

- The following files are not saved by ARCHIVE:
  - $TSOS.TSOSCAT, $TSOS.SYSEAM, $TSOS.PAGINGAREA and $TSOS.SYS.PAGING....
  - disk save files themselves (ARCHIVE.SAVE....) and files which are cataloged under such a name during saving to disk
  - temporary files
  - directory files and print files (ARCHIVE.P....) which are currently being used.
- Parallel processing can be defined by means of the DRIVES operand and the NAME operand of the FILES statement (see page 75).
- The chronological order can be influenced by the definition of a number of FILES or JOBVAR statements.
- As a default for every save run, the LIST operand is used to request a so called report with all the information about the saved files
  (see section “ARCHIVE report” on page 52).
- Encrypted files (BS2000/OSD as of V6.0) will be transferred without change from disk to the save file, for save as well as for migration. Encrypted files are thus stored on the save file in encrypted form.

  A specification of a crypto-password is not required.
  The run time of requests is not affected by encrypted files.

  In HSMS operation, the save of PLAM information for encrypted PLAM libraries will be rejected.
- Tape encryption
  In BS2000/OSD V8.0 or higher and MAREN V12.0 or higher data can be saved to LTO-4 tapes in encrypted format. Only the special volume type TAPE-U4E (see page 93) need be specified in ARCHIVE to use this function.
  For details of tape encryption, see the “MAREN” manual [12].
- See also the section “Smaller request packets for save” on page 102.

5.1.4 Restoring files and job variables (RESTORE statement)

When saved files are to be restored again, only the directory file containing the desired save version need be specified. ARCHIVE then determines the save volume by consulting the directory file and requests it on the console via a MOUNT statement.

5.1.4.1 Save versions

If a directory file contains more than one save version, the file to be restored is taken from the last save version unless otherwise specified.

If the save run was carried out without a directory file, the save volumes must be specified when the files or job variables are restored.

5.1.4.2 Restoring a partial save

If only updated pages (partial save) were saved, the entire file is first restored, i.e. the volumes used for the last full save of the file are requested from the volume pool of the directory file and the file is restored. The volumes of the desired partial saves are then requested and the updated pages are restored.

If you are working with a directory file and specify `FROM=LATEST,STATE` or `FROM=date.time,STATE`, ARCHIVE first writes back the last full save and then the last partial save. The latter contains all the pages which have been modified in the file since the last full save.

If no directory file is being used, the user is obliged to restore in the last full save as well.

5.1.4.3 Output volume

If no output volume is specified for the restore operation, the default volume type is assumed, i.e. the same type as the input volume:

- Files which were saved from a public disk are restored to a public disk.
- Files which were saved from private volumes are restored to the same private volumes (same VSNs).

If files are written back to a volume other than the original one, you must specify the VSN of the new volume in the TO operand of the FILES statement.

When restoring multivolume files on disks of different types, you must reserve storage space with the `/CREATE-FILE` command and execute the restore operation with the `SPACE=KEEP` operand.

Files can be renamed.
In the case of catalog entries saved with `CATONLY=YES`, only the catalog ID may be renamed.

When an entire file is restored, it is assigned a new internal file name (CFID). Such files are not saved in a new save run with CHANGED=YES if they were previously restored from the last save version.

5.1.4.4 File generation groups

The following should be observed when restoring file generations / file generation groups:

- A file generation is restored only if there is an entry for this file in the file generation group (FGG) index. This requires a restoration run with the operand REPLACE=YES.
- When a FGG is restored, the existing generations are completely replaced. The existing FGG index is deleted and restored in accordance with the restored entries.

5.1.4.5 Miscellaneous

- A report can be requested for each restore run with the LIST operand. This report contains file names, version numbers and file sizes etc. Files which could not be restored are also listed with an identifier to this effect.
- For restoration on private disk the VTOC area must be sufficiently large to accommodate all the files to be transferred (cf. the F1SIZE specification for VOLIN in the “Utility Routines” manual [1]).
  One PAM page in the VTOC area is required for between 13 and 15 catalog entries.
- If existing files are to be replaced in a RESTORE run (REPLACE=YES), the user can control via the SPACE operand whether the file is to be
  – deleted before restoration (SPACE=REORG)
  – overwritten in its original physical location (SPACE=KEEP).

  If the file is deleted before restoration, ARCHIVE effectively performs a reorganization of the disk when a disk restore is executed. After the restoration, the files do not occupy the same locations as they did before
  (see the section “Reorganization of disk storage units” below).

  Overwriting files in their original physical locations is important for files which have to occupy a particular physical location, e.g. the center of the disk for files which are accessed frequently.
- If a file which does not yet exist is to occupy a specific physical position on the disk, systems support can create it on the desired disk areas with /CREATE-FILE before the RESTORE run. Precisely these disk extents will then be overwritten in a RESTORE run with SPACE=KEEP.

- Encrypted files (BS2000/OSD as of V6.0) will be transferred without change from disk to the save file, for save as well as for migration. Encrypted files are thus stored on the save file in encrypted form.

  A specification of a crypto-password is not required.
  The run time of requests is not affected by encrypted files.

  In HSMS operation, the save of PLAM information for encrypted PLAM libraries will be rejected.

- When restoring encrypted files (BS2000/OSD as of V6.0) a crypto-password need not be specified. The run time of requests is not affected by encrypted files.

  For encrypted files, the implicit Key/nonkey conversion from the save file is not possible with restore.

5.1.5 Reorganization of disk storage units

ARCHIVE supports the reorganization of disk storage units and the files they contain by means of saving with subsequent restoration. A reorganization is necessary if the files on a disk or a pubset become distributed over more and more extents. Without reorganization, further distribution may be impossible. Moreover, the processing of excessively fragmented files increases the number of input/output operations required.

The system support can implement reorganization with the aid of the following statements:

1. `FILES ...`
   `SAVE CHANGED=NO,...`
2. `FILES ...`
   `RESTORE REPLACE=ALLP,SPACE=REORG,...`

The files can also be deleted in the SAVE run via `SAVE ERASE=ALLP`, or between runs via commands.

You can reorganize pubsets conveniently during ongoing operation using the software product SPACEOPT.

5.2 Transfer of files, job variables and volumes

Files and job variables are saved to tape, MT cartridge or disk by means of the EXPORT statement. After this they can be transferred to another BS2000/OSD system or another user ID by means of the IMPORT statement. The files and job variables can be renamed with the RENAME operand in the FILES or JOBVAR statement.

With the EXPORT statement you can only save complete files from public disk as well as catalog entries of files on tape or MT cartridge. ARCHIVE save data can be copied in an EXPORT run.

Save runs performed using EXPORT exclude save files (`ARCHIVE.SAVE.FILE...`) from saving. This also applies to any other files which may have been cataloged under this name.

The EXPORT statement exports cataloged files and job variables regardless of their BACKUP entry (i.e. also those with `BACKUP=E`).

As of ARCHIVE V2.8A, nonprivileged users may export files and/or job variables which are shareable and readable. Shareable files can only be exported without a directory file. The exported files can subsequently be imported under another user ID. During the export operation, all password and environment attributes of the save file must be reset with the operand `ATTRIBUTES=RESET`.

ARCHIVE does not support protection of individual elements in the PLAM library. Problems can occur as a result if the library is restored using a different user ID.

If no output volume is specified in the IMPORT statement, all files and job variables are restored on public volume sets. The file attributes with which the files are created are described on page 88.

If you want to process private volumes in another BS2000/OSD system, ARCHIVE offers the option of saving only the catalog entries of these volumes in the first system by means of the operand `CATONLY=YES` of the EXPORT statement. In the second system, the catalog entries can be read into TSOSCAT via the IMPORT statement.

For restoration on private disk the VTOC extent must be sufficiently large to accommodate all the files and job variables to be transferred. One PAM page in the VTOC area is required for between 13 and 15 catalog entries.

The replacement of existing files in an IMPORT run (`REPLACE=YES`) can be controlled via the SPACE operand in the same way as was described for the RESTORE run.

5.2.1 File selection options (FILES statement)

Since only complete files can be saved with the EXPORT statement, the selection options are restricted to the following:

- fully or partially qualified file names
- ranges of fully or partially qualified file names
- volumes (VSNs)
- file attributes UNUSED
  (when the file was last accessed for reading or writing).

5.2.2 Transferring files

If files are to be processed in another data center, they can be written to tape, MT cartridge or disk in the first BS2000/OSD system or under the first user ID with an EXPORT run.

The save volumes are then moved to or made available on the second BS2000/OSD system and the files and/or job variables are read in by ARCHIVE with the IMPORT statement. The catalog entries are also constructed in the TSOSCAT catalog.

With ARCHIVE it is also possible to read files or job variables into the new system under another user ID (`SHARE=YES` operand for saving, `RENAME` operand for exporting or importing; `RENAME` is not permitted for exporting with a directory).

ARCHIVE also supports parallel processing if several files and job variables are to be transferred.

The transfer function does not require a directory file. However, it is possible to execute EXPORT/IMPORT runs with a directory file.
If this directory file is the last file to be written to the save volume (using the `DIRSAVE=YES` operand) and is the first to be imported to the second BS2000/OSD system with `DIRECTORY=NONE`, then a table of contents of the files to be imported is made available in the second system. The directory file can then be used to enable the remaining files to be read into the system. This is generally only advisable when you have extensive save files on a number of volumes, which can be processed in parallel runs (DRIVES operand).

5.2.3 Transferring volumes

If files located on private volumes are to be processed in another BS2000/OSD system, the original volumes can also be directly transferred to the second system. The user must then ensure that the catalog entries are created in the catalog TSOSCAT. ARCHIVE offers a convenient solution to this problem with the CATONLY operand:

In the first BS2000/OSD system the catalog entries for the files on the private volume are saved in an EXPORT run which includes the operand `CATONLY=YES`. In the second BS2000/OSD system the catalog entries can be entered in TSOSCAT by way of an IMPORT run. Then the original private disks can be mounted and the files processed.

The transfer of catalog entries is not usually necessary for private disks. Private disks possess their own catalog, which can be transferred to TSOSCAT with `/IMPORT-FILE`.

If the application is limited to the export of catalog entries for volumes, specify `FROM=PRTAPE` in the FILES statement.

5.3 Supporting pubsets

Public disks are combined to volume sets or SF pubsets. One or more volume sets constitute a pubset.
Several pubsets can be installed simultaneously in a system run, thus there are several catalog IDs (CATIDs).

ARCHIVE supports SM pubsets in the same way as SF pubsets. Specific file attributes of SM pubsets are saved and restored. With RESTORE, ARCHIVE automatically selects the pubset that is most suitable and does not require conversion.

In case of a (later) transition to HSMS, it is recommended for SM pubsets that the directories used are restricted to one SM pubset each.

In order to support several catalogs, ARCHIVE offers the CATID operand in the PARAM statement. This operand is used in save and restoration runs:
`PARAM CATID=NO|YES`

This operand can be used to define whether or not ARCHIVE is to save the catalog ID of the saved files and job variables. Its meaning varies depending on whether it is used in a save or restoration run.

The following applies to save runs and restoration runs

Copying of save data must be carried out in the same mode as the original save run. For this reason, either `CATID=YES` or `CATID=NO` must be specified when copying save data.

Saves that were created in an ARCHIVE version < 2.4A, are created with `CATID=NO`

CONTINUE with a mode other than that of the original save is rejected with the message `ARC0277`.

Save runs

In save runs with `CATID=NO` (default value), the catalog ID is ignored. This means that:

- the catalog ID of the saved file is not in the report
- no catalog ID is used for DMS macros; the standard system regulations apply
- space characters are written into the save file in place of the catalog ID.
- the catalog ID is not entered in the FILES records of the directory file
  the attempt to enter a catalog ID in an ARCHIVE statement is rejected
- extending an existing save version (CONTINUE operand) with a different mode to that of the original save operation is illegal.

If `CATID=YES` is specified for a save run, the following applies:

- the catalog ID of the saved file is specified in the report before the user ID in the page header
- files are only sought in the catalogs with the specified catalog ID
- if a file/job variable name is specified without a catalog ID, the standard catalog ID of the relevant user ID is added to the name.

The following changes in the pubset structure during an ARCHIVE run lead to wrong or incomplete results:

- if a pubset which is being processed in the current run is exported
- if the standard catalog ID in the JOIN file of the source pubset is modified for the LOGON user ID or another user ID that is affected by the current run.

The same conditions must be satisfied for restarted ARCHIVE processes as applied before the abnormal termination:

- the standard catalog IDs must not have been modified
- no pubset may be exported or imported between the commencement of the interrupted run and the RESTART run.

Restoration runs

The statement PARAM CATID=NO/YES specifies in which mode the save to be read in was created.

For runs with CATID=NO (default value), the following applies:

- the save run to be read in must have been created with CATID=NO
- the files or job variables are written back to the default pubset of the relevant user ID
- however, they can also be written back to any other pubset specified by means of the RENAME operand
- apart from in the RENAME operand, no catalog IDs can be specified in the ARCHIVE statement
- the report does not contain any catalog IDs, unless one was specified in the RENAME operand, in which case it is always entered in the header line before the user ID.

For restoration runs with CATID=YES the following applies:

- the save run to be read in must have been created with CATID=YES
- by default, the files or job variables are written to the pubset on which they resided at the time of saving (source pubset)
- using the RENAME operand, the files or job variables can be written to any default pubset; in the case of catalog entries, only the catalog ID may be renamed
- this catalog ID is contained in the header line in the report; when RENAME is specified, the catalog ID of the source pubset is contained in the RENAME message.

If a directory file is being used, the catalog ID is transferred to the key of the FILES records in the case of runs with CATID=YES. The directory file contains an identifier indicating whether a catalog ID is present in the FILES records.

During a save run with CATID=YES it is possible to use a directory file created with CATID=NO, provided this directory file is converted before the run with the aid of DIRCONV (see the “HSMS” manual [9]). Following conversion, it is only possible to work with this directory file with CATID=YES.

As a rule, no pubset with the original catalog ID exists for IMPORT runs. When this is the case, the files and job variables must be renamed.

If file names or job variable names are specified without a catalog ID, the standard ID of the user ID in question is taken as the catalog ID. It must be borne in mind that the catalog IDs may be different at the time of the save run and the restoration run.

If the user ID no longer exists, no standard catalog ID can be determined. In this case the catalog ID of the source pubset is added to the catalog ID (for runs with CATID=YES).

5.4 Parallel and serial processing

By default, ARCHIVE processes files and job variables in save and restoration runs serially, i.e. two or more FILES or JOBVAR statements are always processed consecutively.

However, within one FILES or JOBVAR statement, particularly in the case of a system backup, it is possible to write to or read from several save volumes simultaneously (parallel processing).

See also the “Performance measurements in ARCHIVE as of V8.0” on page 102.

Parallel processing

In ARCHIVE, parallel processing is generally important only for the system support. Normally only he/she has to deal with quantities of files large enough to make it worthwhile. However, the option of using parallel processing is available to any user.

Parallel processing in ARCHIVE is controlled by means of the DRIVES operand and, in the case of save runs, also via the FILES statement.

As many parallel runs are started as were specified in the DRIVES operand. Each parallel run is processed by one subtask. These subtasks are each assigned one device, i.e. as many parallel runs may be specified as devices are available. Each parallel run is identified with a subsave number (0,1,...). The subsave number of the first parallel run is 0.

The subtasks receive the names of the files to be processed in “packets” of information from the main task.
Each subtask processes its packet and then requests another. In the case of save runs, which files a packet contains can be controlled by means of the NAME operand in each FILES statement. This takes place as follows:

- In save runs, depending on the distribution of the files to the volumes of the pubset ARCHIVE forms up to four packets for the files to be saved for each user and catalog ID; in addition, the files can be distributed to several packets by means of several NAME operands within a FILES statement.
  The packets within a FILES statement are then processed in parallel by several ARCHIVE subtasks.
- In restoration and import runs as well as when copying save data, the division into packets is performed automatically, based on the way the data was saved to the volumes. Only the same number of subtasks may be active as were used during saving.

It may happen that a run is aborted before all the FILES statements have been executed, e.g. if the operator rejects the tape assignment with `tsn.NO`. In this case, the files which have not yet been transferred and were specified explicitly in the NAME operand are marked `NOT PROCESSED` in the ARCHIVE report.

All that has been said about the FILES statements also applies analogously to the JOBVAR statement.

Example 1

```
/SET-LOGON-PARAMETERS TSOS
/START-ARCHIVE
 . . .
*FILES
*SAVE ...,DRIVES=3
```

Up to four packets are created for each user ID of the system. All the user IDs of the system are saved in parallel on 3 tape devices by 3 subtasks.

Example 2

```
/SET-LOGON-PARAMETERS TSOS
/START-ARCHIVE
 . . .
*FILES NAME=$ABC.
*SAVE DIRECTORY=...,DRIVES=2
```

In this run the files of the $ABC ID are distributed to up to 4 packets and saved to two magnetic tape devices by two subtasks. However, if the pubset consists of just one volume, only one packet is generated. As a result of the `DRIVES=2` operand this means that two subtasks are started, but one of these has nothing to do. Consequently no parallel processing takes place.

Serial processing

The individual FILES statements are processed in strict consecutive order: all files specified in one FILES statement are transferred before the next FILES statement is processed.

The generated subtasks wait if no further packet can be created for them from the first FILES statement.

Example 1

```
/SET-LOGON-PARAMETERS TSOS
/START-ARCHIVE
 . . .
*FILES NAME=$A.
*FILES NAME=$B.
*FILES NAME=$C.
*SAVE DIRECTORY=...,DRIVES=3
```

In this run the three IDs are processed serially. If the files of the various IDs are distributed evenly over the various volumes of the pubset, the three subtasks are utilized evenly. However, if, for example, the pubset consists of only one volume, only one packet can be formed within the FILES statement and two subtasks are in the wait state.

Example 2

```
/SET-LOGON-PARAMETERS TSOS
/START-ARCHIVE
 . . .
*FILES NAME=$A.
*FILES NAME=$B.,NAME=$C.,NAME=($D.AA,THRU=$D.KK9)
*SAVE ....,DRIVES=3
```

In this save run, two subtasks remain in the wait state until the third subtask has completely saved the user ID A. Only when all the files of the user ID A. have been processed do the three subtasks process the files of the second FILES statement in parallel.

When tapes are read in, the whole tape is wound from the beginning for each FILES statement. Users should therefore consider whether or not it is really a good idea to use a number of different FILES statements.

Example 3

```
/SET-LOGON-PARAMETERS TSOS
/START-ARCHIVE
 . . .
*FILES NAME=HUGO1
*FILES NAME=HUGO2
*RESTORE DIRECTORY=NONE,FROM=(RGA001)...
```

This sequence of statements results in tape RGA001 being searched until the file HUGO1 is found. Next the tape is searched a second time for the file HUGO2. In order to find both tapes in a single search, the following sequence of statements must be entered:

```
*FILES NAME=(HUGO1,HUGO2)
*RESTORE DIRECTORY=NONE,FROM=(RGA001)...
```

The FILES statements are processed in strict consecutive order and therefore no coupling of the NAME operands in the individual FILES statements takes place. Files specified in more than one FILES statement (even implicitly) are also processed more than once.

Example 4

If the system support wishes to save the entire public space area with the exception of the user IDs $TSOS and $SYSDUMP, the FILES statement then becomes

```
*FILES EXCEPT=($TSOS.,$SYSDUMP.),FROM=PUBLIC
```

Under $TSOS the missing NAME operand has the effect of implicitly causing all the files in public space to be addressed. In the example the $TSOS and $SYSDUMP. files are excluded by means of the EXCEPT operand. In contrast to this, the following sequence of statements causes all the files in public space to be saved twice, with the exception of the files under the user IDs $TSOS and $SYSDUMP. Both these user IDs would be saved only once.

```
. . .
*FILES EXCEPT=$TSOS.,FROM=PUBLIC
*FILES EXCEPT=$SYSDUMP.,FROM=PUBLIC
*SAVE ...
. . .
```

Summary

- Parallel processing only takes place within a **single** FILES statement, i.e. between the individual user IDs or NAME operands of the FILES statement in the case of SAVE runs.
- A sequence of FILES statements is always processed in strict serial order. In RESTORE and IMPORT runs and when copying save data this means that the input volumes are searched again from the beginning for each FILES statement. Information in the NAME or EXCEPT operand is processed separately for each FILES statement.

5.5 Continuation of save versions

By default, ARCHIVE creates a new save version for each save run. However, it is also possible to extend (continue) an already existing save version, thereby making better use of the volume containing the save data.

The continuation of a save version is initiated via the CONTINUE operand of the EXPORT and SAVE statements. Whenever a save version is continued, the last volume of this version is always requested. The new files or job variables are written after the last file or job variable.

This volume can be specified in the CONTINUE operand in several ways:

- When working with a directory file, it is sufficient to specify the svid. ARCHIVE determines the last volume with the aid of the directory file, and requests it.
- If the VSN is specified (`vsn`,...), ARCHIVE determines the save version once the volume has been read.
  If you specify several VSNs and are using a directory file, ARCHIVE checks whether all volumes belong to the same save version; if not, the run is aborted.
- If the save version and VSNs are entered, ARCHIVE checks the consistency of the entries.

The volumes to be used when the last volume of the save version to be continued is full are specified via the VOLUME operand, i.e. taken by default from the volume pool in the case of a save run. The last volume may not be specified here.

If a directory file is being used, ARCHIVE does not save those files which have already been saved in the save version to be continued or in a later one. ARCHIVE lists these files in the report. This ensures that a file is not contained several times in the same save version, and that the files read in a subsequent RESTORE/IMPORT run reflect the most up-to-date status.

In runs without a directory file, it is the user's responsibility to ensure that the same save version does not contain multiple copies of files. Otherwise it is impossible to tell in a subsequent RESTORE/IMPORT run which version of the file is actually to be restored or imported.
The continuation of save versions is thus not suitable for saving different versions of the same file.

When save versions are continued, their original expiration data is retained.

Example

If the following entries are made, ARCHIVE may save the same file twice, even though a directory is being used.

```
FILES NAME=FILE1
FILES FROM=SV,(MBK001)
SAVE DIR=directory,VOLUME=MBK002,...
```

Reason: if `FILE1` was also on the specified tape (`MBK001`), then it will be saved twice to `MBK002`.
This duplicate saving of files can be avoided with the aid of the CONTINUE operand:

```
FILES NAME=FILE1
SAVE DIR=DIRECTORY,VOLUME=MBK002,...
FILES FROM=SV,(MBK001)
SAVE DIR=directory,CONTINUE=(MBK002),...
```

This statement sequence causes the save data on `MBK002` to be brought up to date: the files on `MBK001` are saved to `MBK002` only if no (more recent) file with the same name has been saved from disk.

The continuation of save versions which have been created with other ARCHIVE versions is not accepted.

5.6 Restarting ARCHIVE processes

Active ARCHIVE processes which require I/O operations on save volumes (i.e. EXPORT, IMPORT, LIST, RESTORE and SAVE) and which have been interrupted (e.g. due to a system crash) can be restarted via the PROCESS statement. Instead of having to begin again from the beginning, they may resume at a defined position during the run.
This is possible only if the ARCHIVE run was started with `PARAM RESTART=YES`. In this case, checkpoints are written into the checkpoint file (`ARCHIVE.CHKPT`), with the help of which the run can be resumed.

Only the commands which were being processed at the time of the interruption are restarted. This means that:

- the command must have been accepted as valid and the save or restore operation must have been started;
- all subsequent commands are ignored and you must enter them again;
- the environment for the restart must be exactly the same as at the time of the interruption (for example, you may not change the pubset or the user ID if a subsequent restart is to be successful).

In order to permit a restart of an ARCHIVE run which consists of several subtasks, of which one is aborted, all other subtasks are terminated immediately.

A restart without a directory file is pointless after an event such as a system crash or a device error. The file being processed prior to the interruption cannot be saved correctly if the EOF markers could not be written in the interrupted save operation.
This file cannot be logged. Message `ARC0060` is issued.

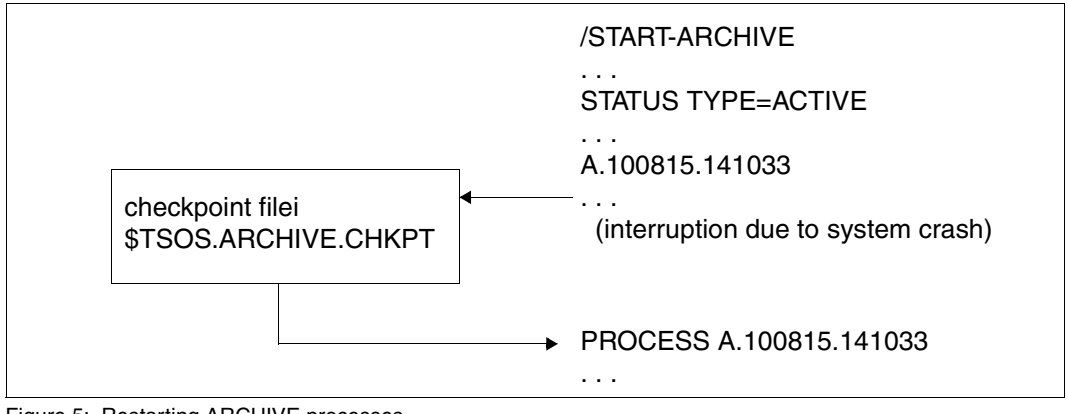


Figure 5: Restarting ARCHIVE processes

5.7 Handling and converting PAM keys

ARCHIVE supports both K disks (with a PAM key) and NK disks (with no PAM key). ARCHIVE saves and restores files with all PAM key formats (BLOCK-CONTROL-INFO=NO/WITHIN-DATA-BLOCK/PAMKEY).

During restoration, files are automatically restored in the format in which they were saved, provided that the format of the volume on which they are restored allows this. The permissible combinations of file attributes and disk types can be found in the table on page 97.

Partial saving is not possible for PAM files without a PAM key (BLOCK-CONTROL-INFO=NO), except in the case of PLAM libraries.

Conversion

ARCHIVE offers the possibility of conversion for cases where the output volume does not permit the writing of PAM keys but the files involved have PAM keys. File conversion can be controlled with the CONVERSION operand of the RESTORE and IMPORT statements. This operand is ignored for job variables.

By default, files are not converted. If the default value is used, files with PAM keys are not restored to an NK disk in a restoration run; a warning is issued for any files which were not restored.

If files are to be converted, the PAMINT subsystem must be available in the file $TSOS.SYSLNK.PAMINT. There are two possible conversion methods:

- CONVERSION=STD
  In a restoration run to an NK disk, files with PAM keys are converted via PAMINT according to the following rules:
  - K-ISAM files to NK-ISAM files
    (BLOCK-CONTROL-INFO=WITHIN-DATA-BLOCK)
  - K-SAM files to NK-SAM files
    (BLOCK-CONTROL-INFO=WITHIN-DATA-BLOCK)
  - K-UPAM files to NK-UPAM files
    (BLOCK-CONTROL-INFO=NO)

    PAM files whose structure is known (load modules, libraries) are converted accordingly; in the case of PAM files whose structure is not known, the PAM key information is lost. If the PAM key contained information, an appropriate message is displayed.

- CONVERSION=CONV
  In a RESTORE run to an NK disk, files with PAM keys are converted to “CONV format” by means of PAMINT. The restored file contains all the PAM keys at the end of the file in separate blocks.

  In the case of partially saved files, only `CONVERSATION=CONV` can be specified.

  If a file to be restored has to be converted, the SPACE operand is ignored.

An SM pubset can contain several volume sets with different format and allocation. When restoring an SM pubset, ARCHIVE searches initially for a “matching” volume set for each file, which does not require conversion. If no such volume set exists, then if possible a volume set will be selected from the SM pubset that allows conversion.

5.8 Data compression

In order to make better use of save volume capacity, you can have ARCHIVE compress the data before writing. Although this increases the load on the CPU, it also reduces the number of I/O operations and the number of save volumes.

Software compression by ARCHIVE is activated with the COMPRESS operand of the EXPORT and SAVE statements. No explicit specification is necessary during data restoration; ARCHIVE automatically decompresses any data compressed in the earlier save run.

Restriction:
Save files that were created with `COMPRESS=YES` cannot be used when installing with the software product SIR.

When saving to device types which perform hardware compression, e.g. tape cartridge devices of the type `TAPE-C4`, no additional software compression is carried out. For these devices, ARCHIVE assumes `COMPRESS=NO`.

The degree of compression varies depending on the saved files; text files, for example, are more tightly compressed than PLAM libraries.
Compression has more influence on the quantity of volumes required than on the time required for saving.

5.9 Online saving of UDS databases

ARCHIVE supports the saving of UDS databases even when they are open. This function is only intended for UDS databases, since it would result in an inconsistent file status if used for saving normal files.

The online saving of UDS databases is possible as of UDS-SQL V1.2 and provided the user is working with an after-image file (for further conditions, see the appropriate manuals for UDS-SQL).

In general, any opened file can be saved online. This requires, however, that the OPNBACK attribute has been set to `YES` with the CATAL(I) program interface and that ARCHIVE is running in `OLS=YES` mode. ARCHIVE does not guarantee the consistency of a file saved in this way.
The online saving of files is controlled via the OLS operand of the PARAM statement.

5.10 Saving the user definition file

ARCHIVE allows the user definition file $TSOS.SYSSRPM to be backed up. This file can however not be saved in itself, but the content can be copied to the file $TSOS.SYSSRPM.BACKUP and can then be saved.

The files $TSOS.SYSSRPM and $TSOS.SYSSRPM.BACKUP exist on every pubset and are managed by SRPM. They are opened on importing the pubset and closed on exporting it.

- The file $TSOS.SYSSRPM with the user definitions cannot be saved, exported or migrated.
- The file $TSOS.SYSSRPM.BACKUP is usually empty and is used to save the file. $TSOS.SYSSRPM. $TSOS.SYSSRPM.BACKUP can however be saved, but not exported or migrated. Only a full save is possible.

  The file $TSOS.SYSSRPM.BACKUP cannot be exported for security reasons. Systems support however can transfer it to another system if necessary using the SAVE/RESTORE functions. With the SAVE/RESTORE functions, ARCHIVE guarantees that only a user with TSOS privilege can restore this security-sensitive file.

Saving the file $TSOS.SYSSRPM.BACKUP

A save can be initiated in ARCHIVE with the SAVE statement or in HSMS with the BACKUP-FILES statement. If an ARCHIVE subtask processes the file $TSOS.SYSSRPM.BACKUP during a save, the following actions are automatically carried out:

1. ARCHIVE notifies SRPM that the save of the user definition file has begun.
2. SRPM copies the data from $TSOS.SYSSRPM to $TSOS.SYSSRPM.BACKUP.
3. ARCHIVE saves the file $TSOS.SYSSRPM.BACKUP on the specified save volume.
4. Once the save of the file $TSOS.SYSSRPM.BACKUP has finished, ARCHIVE informs SRPM.
5. For security reasons, SRPM deletes the data from the file $TSOS.SYSSRPM.BACKUP.

5.11 Restoring and recovering the user definition file

Restoring

The file $TSOS.SYSSRPM.BACKUP cannot be renamed when restoring. Only the catalog ID can be changed. In HSMS, the parameter REPLACE-FILES-AND-JV=*YES or in ARCHIVE, the parameter REPLACE=YES must be specified.
ARCHIVE informs SRPM of the restoration of the file $TSOS.SYSSRPM.BACKUP. SRPM then protects this file against unauthorized access.

Recovering

The file $TSOS.SYSSRPM.BACKUP must be restored before /SHUTDOWN or /EXPORT-PUBSET. On startup or /IMPORT-PUBSET, the message NSI6005 must be responded to with the recovery type or in the case of a FAST startup, the system parameter RECONUC must have the correct value:

- **RECONUC=N**
  The user definition file is not recovered.
- **RECONUC=B**
  A new user definition file $TSOS.SYSSRPM is created. It contains the content of the file $TSOS.SYSSRPM.BACKUP.
  Warning: All files and job variables belonging to users that are not present in the file $TSOS.SYSSRPM will be lost.
- **RECONUC=T**
  A new user definition file $TSOS.SYSSRPM is created, with all users that have files or job variables in the file $TSOS.TSOSCAT. The user definition, if available, will be taken from the $TSOS.SYSSRPM.BACKUP file, or the user will be created with the default values.
- **RECONUC=A**
  A new user definition file $TSOS.SYSSRPM is created, with all users who have files or job variables either in the $TSOS.TSOSCAT file or are entered in the $TSOS.SYSSRPM.BACKUP file. The user definition, if available, will be taken from the $TSOS.SYSSRPM.BACKUP file, or the user will be created with the default values.
- **RECONUC=R**
  A new user definition file $TSOS.SYSSRPM is created, with all users who have files or job variables in the $TSOS.TSOSCAT file. All users are created with default values and are locked, with the exception of users with the TSOS privilege.

5.12 File attribute handling

ARCHIVE saves not only the data part of files and job variables, but also the metadata part (catalog entry) containing the attributes. These attributes can be divided into four groups:

- **Password attributes**

  These are defined using the following operands in the /CREATE-FILE and /MODIFY-FILE-ATTRIBUTES commands:

  IO-ATTRIBUTES, DISK-WRITE, WRITE-PASSWORD, READ-PASSWORD, EXEC-PASSWORD, SPECIFY-ACCESS, DESTROY-BY-DELETE, AUDIT, SPACE-RELEASE-LOCK, BACKUP-CLASS, SAVED-PAGES and MIGRATE

  In the macro, the following operands are used in addition:
  ENCRYPT, FTINFO, NTBEIMC and OPNBACK

  All attributes processed with COPY-FILE PROTECTION=SAME are processed by ARCHIVE as well.

- **Attributes for access protection**

  These attributes relate to file protection and are defined by the owner environment (user ID and catalog ID):

  GUARDS, BACL, ACCESS and USER-ACCESS

  The attributes for access protection are handled likewise via the command /COPY-FILE PROTECTION=SAME.

- **Date attributes**

  These include the creation date, expiration date and last-access date.

- **Other file attributes**

  These relate to the contents of the data part (e.g. FCBTYPE, BLKCTRL, ...).

Handling of these four attribute groups depends on the desired application (backup or transfer):

- **Backup**

  This application is effected by SAVE/RESTORE runs without any change of the owner environment, i.e. the catalog ID and user ID stay the same.
  For backup, ARCHIVE attempts to retain the attributes valid at the time of saving.

- **Transfer**

  This application is effected by EXPORT/IMPORT runs or SAVE/RESTORE runs with a change of the environment, i.e. the catalog ID and/or user ID are modified.
  For transfer, ARCHIVE attempts to reset the password, access protection and date attributes and to retain the other file attributes.

How ARCHIVE handles the attributes in accordance with the considerations mentioned above and with existing compatibility restrictions is described below.

- If a file is not protected with GUARDS or BACL, the *access protection and password attributes* for RESTORE are reset to the values they had when the file was saved.

  In the case of an EXPORT run with `ATTRIBUTES=RESET`, the *access protection and password attributes* in the save file are reset; during a subsequent IMPORT, they are then set to match the status of the disk.
  In an IMPORT run, the *access protection and password attributes* are either reset to the DMS default values (if the file to be replaced does not exist) or they are set to the same values as the existing file.

- If a file is protected with GUARDS or BACL, handling of the *attributes for access protection* depends on the type of application:

  – For a *backup* (i.e. without renaming the catalog ID or user ID), all attributes for access protection are reset to the values they had when the file was saved.

    For restoring to a private disk, the attributes for access protection are reset.

  – For a *transfer*, the attributes for access protection are either reset to the DMS default values (if the file to be replaced does not exist) or they are set to the same values as the existing file.

    Exception
    The operand `ENVIRONMENT-ATTRIBUTES=FROM-ORIGIN` of the RESTORE statement permits a file to be restored while renaming the user IDs. The system support may rename both the catalog ID and the user ID, the nonprivileged user only the catalog ID. The attributes for access protection are then reset to the values they had when the file was saved.

- If a file is protected with GUARDS or BACL, the *password attributes* always retain the values they had at the time of saving.
- The *date attributes* are retained for a backup, but are reset for a transfer.
- In save runs with SAVE-ACL=NO, the access control list is not saved on the tape. In this case, the behavior of the *attributes for access protection* described above applies.
- The *other file attributes* are retained: the data part is not affected by a change of environment or by the type of application.

The two following tables summarize the values assumed by the attributes in a RESTORE or IMPORT run:

1. Attributes of files/job variables which are **not** protected with GUARDS or BACL in the case of SAVE/EXPORT or save file created via EXPORT with the operand ATTRIBUTES=RESET.

	Save version processed with RESTORE	**Save version processed with IMPORT**	
		Is there a file with the same name on the disk?	
		Yes	**No**
Values of the password attributes	same as for save file	same as for save file or disk file [1]	same as for save file
Value of ACCESS USER-ACCESS, BACL and GUARDS	same as for save file	same as for save file or disk file [1]	same as for save file
Values for last-access, creation and expiry dates	same as for save file	current date	current date

[1] Save file created via EXPORT with the operand ATTRIBUTES=RESET

2. Attributes of files/job variables which are protected with GUARDS or BACL in the case of SAVE/EXPORT; the save file was not created via EXPORT with the `ATTRIBUTES=RESET` operand.

	Save version created with SAVE, without renaming the catalog ID and user ID	
	File is restored to	
	PUBLIC	**private disk** (saved from PUBLIC)
Value of the password attributes	same as for save file	same as for save file
Value of USER-ACCESS	same as for save file	OWNER-ONLY
Value of ACCESS	same as for save file	WRITE
Value of BACL and GUARDS	same as for save file	OFF
Values for creation, expiration and last-access dates	same as for save file	same as for save file

	Save version created with EXPORT or SAVE (with renamed catalog ID or user ID in the case of SAVE)		
	Is there a file with the same name on the disk?		RESTORE with ENVIRONMENT-ATTRIBUTES= FROM-ORIGIN
	Yes	**No**	
Value of the password attributes	same as for save file	same as for save file	same as for save file
Value of USER-ACCESS	same as for disk	OWNER-ONLY	same as for save file
Value of ACCESS	same as for disk	WRITE	same as for save file
Value of BACL and GUARDS	same as for disk	OFF	same as for save file
Values for creation, expiration and last-access dates	current date	current date	same as for save file

5.12.1 BACL and GUARDS

The access protection mechanism BACL is an integral component of BS2000/OSD-BC. ARCHIVE saves the BACL attributes together with every file.

The SECOS functional unit GUARDS can be used as a data access protection mechanism for files, libraries and library elements, job variables and FITC ports.

GUARDS represents an independent object management (analogous to DMS or LMS) which handles its objects, the GUARDs. These GUARDs store conditions (but not access rights) that are evaluated on a request from another object management (DMS, LMS, FITC).

The GUARDs are managed by GUARDS. Only the name of the GUARD to be used for protection is stored as a reference under the relevant object management. The object management in question is responsible for assigning the result of the conditions evaluated by GUARDS to an access right.

The file catalog entry merely contains the name of the GUARD for the corresponding access right; the conditions (or contents) remain under GUARDS management. This is why only the references can be entered during restoration. The GUARD contained in the catalog entry is unaffected by restoration.

ARCHIVE behaves in the following ways, depending whether or not a file exists for restoration:

- If the file exists, the current protection attributes are retained; only the file contents are exchanged.
- If the file does not exist, the file is restored with the protection attributes valid at the time of saving.

After each restoration, a check must be made to determine whether the GUARDs designated for protection still contain the desired conditions or whether they have been changed by the GUARD owner since the file was saved.

For details on GUARDS see the “SECOS” manual [14].

5.13 Handling magnetic tapes and magnetic tape cartridges

ARCHIVE can only process magnetic tapes and MT cartridges that have been created by ARCHIVE.

ARCHIVE supports magnetic disks and all TAPE class volumes supported by BS2000/OSD as of V6.0. Details of the device and volume types are provided in the “System Installation” manual [6].

ARCHIVE supports direct positioning when reading MT cartridges. This enhances performance when copying, restoring and listing save files.

5.13.1 Requesting devices

Under the user task (main task), the ARCHIVE user can reserve magnetic tape devices and MT cartridge devices for subtasks with:

```
/SECURE-RESOURCE-ALLOCATION
```

by entering the logical device-type code, e.g. for magnetic tape and MT cartridge:

```
/SECURE-RESOURCE-ALLOCATION DEVICE=*PAR(TYPE=TAPE-C4,NUMBER=3)
```

or in conjunction with the mnemonic device name

```
/SECURE-RESOURCE-ALLOCATION UNIT=mn
```

(For further details see the “Commands” manual [4].)

However, no operands may be specified which effect an implicit assignment of the device, i.e. for `/SECURE-RESOURCE-ALLOCATION`, the TAPE and FILE operands are not permitted.

If reserved tape units or MT cartridge devices are available, the main task supplies a free device of the appropriate type and of the appropriate assignment for each subtask. ARCHIVE assumes that all volumes to be processed have the same assignments as the VSNs specified for `/SECURE-RESOURCE-ALLOCATION`.

The device type for `/SECURE-RESOURCE-ALLOCATION` must match the DEVICE operand specified in the EXPORT or SAVE statement, in order that the subtask can access the device.
If a storage location other than the default MAREN storage location is being used when working with ARCHIVE, the MAREN storage location must be specified both in the `/SECURE-RESOURCE-ALLOCATION` command and in the ARCHIVE statements EXPORT and SAVE (MAREN-LOCATION parameter).
No MAREN storage location can be specified in the IMPORT and RESTORE statements. In this case, the MAREN storage location must be specified in `/SECURE-RESOURCE-ALLOCATION`.

Once the subtask has been terminated, the right to access the reserved tape units or MT cartridge devices is returned to the main task. When copying save data, requests for tapes or MT cartridges are also processed in the same way.

When copying saved data, requests for tapes and MT cartridges are handled as described above.

Files that are saved or restored and save files may not be reserved using the `/SECURE-RESOURCE-ALLOCATION` command.

When using HSMS, the devices cannot be reserved, since the reservation cannot be passed to the HSMS server task. The reserved devices would then not be available to an ARCHIVE task.

5.13.2 Error handling

S**AVE/EXPORT runs**

- Write errors
  In the event of an unrecoverable error when writing to a tape or MT cartridge, ARCHIVE issues message `ARC0701` at the console and on SYSOUT. ARCHIVE then attempts to continue the save run without data loss or skips the errored file and continues the save run.
  If, however, the end-of-file identifiers cannot be written, the entire save run is aborted. An error-free restoration run cannot be performed with such a tape or MT cartridge.
- Read errors
  If an error occurs during a tape-to-tape save (copying of save tapes or MT cartridges) when the input tape is being read in, the copying of the file and the run are continued. The error is reported by means of appropriate messages at the console and on SYSOUT.

RESTORE/IMPORT runs

The handling of read or format errors and open errors varies in accordance with specifications in the OPERATOR operand of the PARAM statement.

`OPERATOR=NO` In the event of a read or format error, ARCHIVE implements default error handling.

`OPERATOR=YES` A message is output at the console in the event of a read or format error during restoration.
The following responses are possible:

`tsn.0`
ARCHIVE implements default error handling (standard routine).

`tsn.1`
ARCHIVE implements a special error handling routine.

Open errors (tape labels cannot be read)

- Standard routine
  If `OPERATOR=NO` was specified, the tape is skipped and the run resumed with the next tape. Any file which could not be restored completely bears an identification to this effect in the ARCHIVE report.
- Special routine
  If the response `tsn.1` is given to the message `ARC0903`, the system tries to open the magnetic tape or MT cartridge again, although this time with `LABEL=NO` (overriding the volume label).
  - If this is possible, the restoration run is continued.
  - If this is not possible, the tape or MT cartridge is skipped. One or more messages for device errors are then output on the console, and the operator must respond to these with `tsn.1`.

  
  This method of error recovery is, for data security reasons, only possible under the user ID of systems support. `TPIGNORE=YES` must be set in the user catalog.

Read or format errors during restoration

- Standard routine
  Restoration of the file to be written back is aborted and the file in question is marked appropriately in the ARCHIVE report. The ARCHIVE run is continued from the next file which can be read.
  Partially saved files are always handled in this way.

- Special error handling routine
  If `OPERATOR=YES` is specified in the PARAM statement, and if the messages `ARC0901` (read error) and `ARC0902` (format error or incorrect block number) are responded to with `tsn.1`, then ARCHIVE resumes restoration of the appropriate file.

  If the message `ARC0906` (incorrect block number) is responded to with `tsn.1`, ARCHIVE continues processing of the appropriate tape. If read or format errors occur, then ARCHIVE enters a dummy block in the first PAM page of the series of PAM pages which cannot be read. The following text appears at the start of the dummy block:

  `*****ARCHIVE RESTORE DUMMY BLOCK*****`

  – In the case of files with PAM keys (`BLOCK-CONTROL-INFO=PAMKEY`), the other PAM pages of this series are not written and therefore have a different internal file name (coded file ID, CFID) from the restored file.

  – In the case of files without PAM keys (`BLOCK-CONTROL-INFO=NO/WITHIN-DATA-BLOCK`) each unreadable PAM page is replaced by a DUMMY block.

  Restoration of the file then continues from the next readable block.

An invalid block number causes all PAM pages to be restored from the save file. The printout of the ARCHIVE save file concerning the environment of the particular block lists any PAM pages that are missing from the file.

If more than 99 read or format errors occur on a tape or a MT cartridge, reading of this tape or MT cartridge is aborted. The run is resumed with the next input tape or MT cartridge.

A format error of a special type occurs when the ARCHIVE header block of a tape or MT cartridge is faulty. In this case, ARCHIVE rejects the tape or MT cartridge with message `ARC0249`. This is, for example, the case with tapes or MT cartridges created as continuation tapes or MT cartridges when duplicating an ARCHIVE tape or MT cartridge with a product other than ARCHIVE.

5.14 Disk handling

This section deals with private disks, NK4 disks and error handling during disk processing.

5.14.1 Private disks

For private disks that are to be accessed by ARCHIVE, you must enter the following command:

```
/SET-DISK-PARAMETER ...,USER-ALLOCATION=*SHARE,ASSIGN-TIME=*OPERATOR
```

A file on several private disks with different device types can be saved and restored as well (ARCHIVE and HSMS operation).

5.14.2 NK4 disks

ARCHIVE enables you to use NK4 disks. The following are supported:

- the operands BLOCK-CONTROL-INFO and BUFFER-LENGTH of ARCHIVE *work files*. As of ARCHIVE V2.8A, there are no restrictions for the operand BUFFER-LENGTH of the directory file.
- the operands BLOCK-CONTROL-INFO and BUFFER-LENGTH of ARCHIVE *disk save files*.
- saving from and restoring to NK4 disks.

  Save runs also process NK4-ISAM files. For such files, the control field in each logical block has two PAM pages.

  The following combinations are possible for restore runs:

Attributes of the saved file	**RESTORE to**		
	K disk	**NK2 disk**	**NK4 disk**
BLOCK-CONTROL-INFO = PAMKEY BUFFER-LENGTH = (STD,n) (n odd)	1)	2)	4)
BLOCK-CONTROL-INFO = PAMKEY BUFFER-LENGTH = (STD,n) (n even)	1)	2)	2)
BLOCK-CONTROL-INFO = PAMKEY BUFFER-LENGTH = (STD,n) (n even) PAM phase	1)	6)	6)

Attributes of the saved file	RESTORE to		
	K disk	NK2 disk	NK4 disk
BLOCK-CONTROL-INFO = NO/ WITHIN-DATA-BLOCK BUFFER-LENGTH = (STD,n) (n odd) not ISAM	1)	1)	3)
BLOCK-CONTROL-INFO = NO/ WITHIN-DATA-BLOCK BUFFER-LENGTH = (STD,n) (n even) not ISAM	1)	1)	1)
BLOCK-CONTROL-INFO = WITHIN-DATA-2K-BLOCK BUFFER-LENGTH = (STD,n) (n odd) ISAM	1)	1)	4)
BLOCK-CONTROL-INFO = WITHIN-DATA-2K-BLOCK BUFFER-LENGTH = (STD,n) (n even) ISAM	1)	1)	5)
BLOCK-CONTROL-INFO= WITHIN-DATA-4K-BLOCK BUFFER-LENGTH = (STD,n) (n even) ISAM	1)	1)	1)

1) The file is restored with the attributes which it had when it was saved. No conversion is necessary.

2) If there are no incompatibilities (record length, ...), you can carry out a standard conversion (CONVERSION=STD) with PAMINT during the restore operation. This sets the file to BLOCK-CONTROL-INFO=WITHIN-DATA-BLOCK (SAM/ISAM) or to BLOCK-CONTROL-INFO=NO (PAM); the value for BUFFER-LENGTH remains unchanged.

   If there are incompatibilities, you can convert the file into the CONV format (CONVERSION=CONV) with PAMINT during the restore operation. This makes the file a PAM file with BUFFER-LENGTH=(STD,2).

   In both cases, the file can be restored.

3) The file cannot be restored. The value for BUFFER-LENGTH must be changed. You must therefore use the PAMCONV program prior to saving so that this file can be restored to an NK4 disk.
   You may also convert this file into the CONV format with PAMINT, but a file converted in this manner cannot be used by the access method SAM/ISAM.

4) The file cannot be restored. Both the value for BUFFER-LENGTH and the value for BLOCK-CONTROL-INFO (WITHIN-DATA-BLOCK for SAM, NO for PAM, WITHIN-DATA-4K-BLOCK for ISAM) must be changed. You must therefore use the PAMCONV program prior to saving so that this file can be restored to an NK4 disk. You may also convert this file into the CONV format with PAMINT, but a file converted in this manner cannot be used by the access method SAM/ISAM.

5) The file cannot be restored. The value for BLOCK-CONTROL-INFO must be changed from WITHIN-DATA-2K-BLOCK to WITHIN-DATA-4K-BLOCK.
You must therefore use the PAMCONV program prior to saving so that this file can be restored to an NK4 disk.
You may also convert this file into the CONV format with PAMINT, but a file converted in this manner cannot be used by the access method SAM/ISAM.

6) The file cannot be restored. You must therefore use the PAMCONV program prior to saving so that this file can be restored to an NK4 disk.
You may also convert this file into the CONV format with PAMINT, but a file converted in this manner cannot be used by the access method PAM.

5.14.3 Error handling for disk processing

EXPORT/SAVE runs

- Write errors

  In the event of an error when writing to disk, message ARC0702 is issued and the save run is aborted.

- Read errors

  If a read error occurs during the copying of disk save files, restoration of the current file is terminated and restoration of the next file commences.

5.15 Alias names for files

Alias names (see the “DMS” manual [2]) are permitted in ARCHIVE. An alias name is virtually any name which the user may use instead of the actual file name. The relationships between the alias names and the file names are defined in the alias catalog.

An alias name is accepted wherever a file name is selected with the `FILES NAME=` or `INQUIRE FILES=` statement or with the DIRECTORY operand.

Internally, ARCHIVE uses only file names, not alias names. For this reason, all messages and listings contain only file names. Moreover, the ARCHIVE statement is rejected with message `ARC0252` in `CATID=NO` mode, if ACS converts an alias name specified by the user into a file name with a catalog ID.

If an alias name is selected in an EXPORT or SAVE statement, the save file is addressed with its file name.
If an alias name is selected in a LIST, IMPORT, INQUIRE or RESTORE statement, the corresponding file name is used to search the work file (save file or directory file).
If no file is explicitly selected (e.g. FILES statement without the NAME operand), the file name is processed directly. For this reason, it is not possible to implicitly rename a file with the alias mechanism during an IMPORT or RESTORE operation.

The user should avoid the ACS entry `COMPLETE-ALIAS-NAMES=ALLOWED` because if complete alias names (with catalog ID and user ID) are allowed, multiple file name replacements may occur which will lead to unpredictable ARCHIVE results.

5.16 Duplicate files/job variables

During a SAVE or EXPORT run, the files or job variables may come both from an input tape and from a disk.

The DUPLICATE operand, which may be specified in the SAVE and EXPORT statements, supports files/job variables which exist on both tape **and** disk, in order to obtain a precise status of the file. The first file/job variable referred to in the FILES/JOBVAR statement is processed.

By specifying `DUPLICATE=NO`, it is possible to ensure that files/job variables are processed only once, and are thus saved/exported only once. A directory file is necessary for the save or export run in this case.

Checking for duplicate files increases the runtime of ARCHIVE.

5.17 Performance measurements

- For ISAM access to ISAM files in ARCHIVE (directory, checkpoint, report files), ARCHIVE uses the task-independent standard ISAM pool for data caching. The size of the standard ISAM pool can be set with the system parameter `GLBDFPS`, see “Introductory Guide to Systems” [3].
  Default value: 512 PAM pages (i.e. 1 Mbyte); this value should be increased in accordance with the size of the directory file. Recommended value: 8.192 PAM pages.
- The runtime of long ARCHIVE runs can be reduced using parallel runs on different machines, see DRIVES operand in the EXPORT and SAVE statements.
- Disk processing has been improved in ARCHIVE V6.0 by increasing the standard block size of save files on disks. The block size is now 64 Kbyte. This block size must not be reduced between an EXPORT/SAVE statement and a RESTORE/IMPORT statement.

Performance measurements in ARCHIVE V9.0

Without change of function or usage, some measurements were implemented for better performance, esp. for shorter save runs.

- For D3435-Disks with nonkey, the maximum IO size was increased to 240 PAM pages. The Page-Fixing of the IO buffers will be processed now only once per request, not for each IO. Both improvements are used in BS2000/OSD as of V8.0.
- Loss of performance during save of many small files has been reduced.
- The reconstruction of a directory from tape - saved with the DIRSAVE function - has been speeded up, so that the size or capacity of the tape has practically no impact. A directory is found on the tape not by sequential reading, but by exact positioning from the end-of-tape label.

Performance measurements in ARCHIVE as of V8.0

- Better performance by usage of PAV

  The EMC^2 disk storage systems offer parallel access volumes (PAV), so that parallel disk IOs provide higher throughput. BS2000/OSD as of V6.0 uses this function by starting several IOs for a disk in parallel (see the “Introductory Guide to System Support” [3]).

  ARCHIVE exploits this function for higher throughput during save/restore processing and also with writing to or reading from disk save files.

  This function for better disk performance provides shorter overall times for save and restore, since modern MBK devices have higher data rates than disks and any improvement of the disk side gives overall improvement.

- Smaller request packets for save

  Files to be saved are distributed to the ARCHIVE subtasks in smaller internal packets for a smoother load balancing of the parallel subtasks and a faster overall save processing (for SAVE and EXPORT).
  The files of one catalog ID and one user ID are split into up to four packets depending on their disk allocation (before ARCHIVE V8.0 all grouped in one packet). The packets are independantly distributed to free subtasks for save processing.
  This allows parallel tape processing even in case of a save for just one user ID.

Performance measurements in ARCHIVE as of V7.0

- Performance of copy processing

  The copy of save versions or save files without extra file selection is done via a fast path with minimal access to the input directory.

- New tape format for big tape blocks

  For optimal tape performance and increased volume capacity a new tape format has been introduced, which uses tape blocks of size 256 KByte on MTC as of TAPE-C3 and on BS2000/OSD as of V6.0, see description of the parameter “DEVICE-TAPE-T-C” on page 197 and the ARCHIVE statements SAVE and EXPORT.

  After complete migration to BS2000/OSD V6.0, you should use the new tape format, also for existing archives or directories. The new tape format should not be used when the data also has to be restored in BS2000/OSD < V6.0 (e.g. in the case of mixed mode of various BS2000/OSD versions or data exchange with other data centers).

6 ARCHIVE statements

This chapter first describes the notational conventions used for the ARCHIVE statements. This is followed by an explanation of how to enter and continue statements. The ARCHIVE statements are then summarized in the form of a table, followed by descriptions of the individual statements in alphabetical order.

6.1 Metasyntax

Appears as	Meaning	Example
UPPER-CASE-LETTERS or special characters	Uppercase letters and special characters denote keywords or constants and must be entered by the user exactly as shown.	`NOW=`

lower-case-letters	Lowercase letters denote variables which, on entry, must be replaced by current values.	`DIRECTORY=filename`
		possible entry:
		`DIRECTORY=ARCHIVE.DIR`

{ }	(Complex) entries in braces are alternative, i.e. you must select one of the options.	`DIR={filename/NONE}`
		possible entries:
		`DIR=SAVE` or `DIR=NONE`

/	The slash denotes a choice between alternatives; it has the same function as braces.	`NOW=YES/NO`
		possible entries:
		`NOW=YES` or `NOW=NO`

[...]	Square brackets enclose optional specifications, i.e entries that can be omitted. When a comma is enclosed between square brackets in optional entries, it need only be typed if the option is used. When it is outside the brackets it must be specified even if the option is not used (round brackets must be entered).	`YES[,LARGE/PAM pages]`
		
		possible entries:
		`YES` or `YES,LARGE`
		or `YES,5`

Appears as	Meaning	Example
Underlining	Underlining highlights a default value. For parameters that are defined in the ARCHIVE parameter file, the presetting is entered (cf. page 194). The value for the parameters can differ from the presetting if the parameter file has been changed (cf. page 23). For other parameters, the default value is the value that ARCHIVE inserts if you do not make an entry. If no default value is specified, you must make an entry.	NOW=YES/NO possible entries: NOW=YES or NOW=NO or no entry, then the value is taken from the ARCHIVE parameter file. DRIVES=1/integer possible entries: DRIVES=2 or no entry, then DRIVES=1 is used. PASSWORD=password
...	Ellipses denote repetition, i.e. the preceding syntactical unit may be specified one or more times in succession.	(vsn,...) possible entries: (C5432A,C5003A) or (C5432A,C5003A,C5112A) etc.
=	An equals sign links the operand name with the operand values associated with it.	NOW=YES/NO

6.2 Data types

Representation	Explanation	Format
alphanum- name 1..8	Name (e.g. of the MAREN storage location)	A...Z, 0...9, $, #, @ Length: 1 to 8 characters
archive-seqno	Identifier linked to a statement processed by ARCHIVE. Used in the DELETE and PROCESS statements.	– A.yymmdd.hhmmss for an active statement – L.yymmdd.hhmmss for statement sequences executed later (operand NOW=NO) – S.yymmdd.hhmmss for completed ARCHIVE runs, where S is entered in the directory file as a save version identifier yy: year, mm: month, dd: day, hh: hour, mm: minute, ss: second
backup	Only those files are saved for which the catalog holds a BACKUP entry equal to or less than this value (A < B < C < D).	A, B, C, D
catid	Catalog ID. The catalog ID entry must be enclosed in colons (:catid:)	A..Z, 0..9 Length: 1 to 4 characters; as defined in the “DMS” manual [2]
filename	Partly or fully qualified file name; an alias is also permitted. The catalog ID can also be specified for runs with PARAM CATID=YES (NAME=:c:$userid.filename). The catalog ID can also be entered alone.	Length: 1 to 54 characters; (including catalog ID and user ID); as defined in the “DMS” manual [2]
date	Date specification with year (yy or yyyy), month (mm) and day (dd); entries with yy for year are internally converted to yyyy, the year 1960 being assumed as the limit: – 771201 is interpreted as 19771201 – 601005 is interpreted as 19601005 – 512801 is interpreted as 20512801	yymmdd or yyyymmdd
device type	device type for disk device, tape device or MT cartridge device.	As defined in the “DMS” manual [2]

Representation	Explanation	Format
jv-name	Partly or fully qualified name of a job variable. Under the TSOS ID, a foreign user ID (`$userid.jv-name`) can be entered. For runs with `PARAM CATID=YES`, a catalog ID can also be entered (`NAME=:c:$userid.jv-name`).	char 1..54 (including catalog ID and user ID); see the “DMS” manual [2] for details
password	Read, write or execute password	– `C'cccc'` (up to 4 chars.) or – `X' hhhhhhhh'` (up to 8 hexadecimal digits) or – `<integer -2147483648 ..2147483647>` see the “DMS” manual [2] (PASSWORD command) for details
PAM pages sec days integer	Integer, the permitted value range is more precisely defined in the operand description if necessary.	`0..9`
svid	Save version identifier	`S.yymmdd.hhmmss` `yy`: year, `mm`: month, `dd`: day, `hh`: hour, `mm`: minute, `ss`: second
userid	User ID	`A..Z, 0..9, $, #, @` Length: up to 8 characters; as defined in the “DMS” manual [2]
vsn	Volume serial number of a magnetic tape, MT cartridge or disk; assigned to the volume on initialization (VOLIN or INIT).	Length: 6 characters; as defined in the “DMS” manual [2]
time	Time specification in hours (`hh`), minutes (`mm`) and seconds (`ss`)	`hhmmss`

6.3 Statement input format

Your input to ARCHIVE may consist of several statements, which must be separated from each other by at least one blank. The total length of the input may not exceed 2044 characters.

When defining a save or restoration run the statements must be entered in the following order:

1. a PARAM statement (optional)
2. one or more FILES and/or JOBVAR statements
3. a statement involving I/O operations on save volumes (e.g. SAVE or RESTORE)

This series of statements is organized into an internal representation by ARCHIVE. If this causes the input area to be exceeded, message `ARC0121` is issued.

If an error is made when entering the statements, the FILES and JOBVAR statements must also be reentered when the correction is made.

Example 1

```
FILES NA=FAR.E   EXPORT NOW=YES,TAPES=WORK01
1st statement    2nd statement
```

Each statement begins with the statement name (e.g. FILES). This must be followed by at least one blank. The operands must be separated by commas. You may omit the comma after a right parenthesis except before the DEVICE operand in a FILES statement.

Example 2

```
EXPORT      NOW=YES,TAPES=WORK01
Operation   Operands
```

Each line entered may contain extra blanks, in any position except in file names or keywords (e.g. `TAPES`).

The operands may be in any order.

The specifications `YES` and `NO` can always be shortened to Y and N.

Continuation of statements

In interactive mode, input can be continued on the next line without the need for a continuation character.
If SYSDTA is not allocated to a terminal (but to a SAM or ISAM file or a floppy disk), any statement may be continued in the next line if the line to be continued ends with one of the following characters:

- a hyphen “-” in column 72
- a comma “,” or an equals sign “=” as the last character in an input line
- a right parenthesis “)” as the last character in an input line of a FILES or JOBVAR statement.

A comment is written in quotes (" "). Like the statement characters, the inverted commas come before column 73.
You can also include comments in statements wherever blanks are permitted. A comment can comprise several lines.

If SYSDTA is not assigned to a terminal, all characters in columns 73-80 are ignored.

6.4 Overview of the ARCHIVE statements

Statement	Function
DELETE	Deletes waiting ARCHIVE processes from the queue file ARCHIVE.LATER
END	Terminates the ARCHIVE run
EXPORT	Saves job variables, catalog entries and complete files to tape, MT cartridge or disk, from where they can be transferred to another BS2000/OSD system or another user ID in a subsequent IMPORT run
FILES	Defines the files to be saved or read in, together with the related volumes
HELP	Lists ARCHIVE statements with possible operands and a brief explanation in German or English
IMPORT	Reads files, job variables or catalog entries saved in an EXPORT run from tape, MT cartridge or disk into another BS2000/OSD system or another user ID
INQUIRE	Displays information about a directory file
JOBVAR	Defines the job variables to be saved or read in
LIST	Creates a list of files, job variables and catalog entries saved on tape, MT cartridge or disk
PARAM	Sets parameters for the entire ARCHIVE run
POOL	Adds tapes or MT cartridges to the pool of the directory file or removes them from this pool
PROCESS	Starts waiting ARCHIVE processes from the queue file ARCHIVE.LATER or continues interrupted processes
PURGE	Deletes save versions from the directory file
RESTORE	Restores files, file blocks, job variables or catalog entries saved on tape, MT cartridge or disk in a SAVE run and writes them to public or private volumes or to the system catalog
SAVE	Saves files, modified file blocks, catalog entries and job variables to tape, MT cartridge or disk, from where they can be restored in a subsequent RESTORE run
SHOW-DEFAULT	Shows the current default values for the ARCHIVE parameters.
STATUS	Informs the user whether an ARCHIVE process is active, can be started or is in the queue file ARCHIVE.LATER

6.5 Description of the ARCHIVE statements

The description of each ARCHIVE statement begins with a general text which explains the function of the statement. This is followed by the statement format and a description of each operand and the related values.

DELETE
Delete ARCHIVE processes

The following types of ARCHIVE processes can be canceled by means of the DELETE statement:

- processes defined with the operand NOW=NO which have not yet been started;
- processes which have been interrupted.

DEL[ETE]
archive-seqno

archive-seqno
15-digit ARCHIVE number in the form `A.yymmdd.hhmmss` or `L.yymmdd.hhmmss` for running or waiting processes respectively (yy=year, mm=month, dd=day, hh=hour, mm=minute, ss=second).

The system support may omit this operand. If this operand is not specified, all the ARCHIVE processes defined with the operand `NOW=NO`, which are thus in the queue file `ARCHIVE.LATER`, are deleted.

Example

See page 32.

END
Terminate ARCHIVE

The END statement terminates the program ARCHIVE.

END

EXPORT
Save to volume for transfer

The EXPORT statement enables files, catalog entries or job variables to be saved to tape, MT cartridge or disk for subsequent transfer to another BS2000/OSD system or to another user ID by means of an IMPORT run. In addition, save data created in an earlier EXPORT run can be copied. The save files created via EXPORT can be read in again by means of an IMPORT run.

Before you enter the EXPORT statement, you must define the files and/or job variables to be processed by means of FILES and/or JOBVAR statements.

The EXPORT statement saves all files, regardless of the BACKUP entry.

E[XPORT]

DIR[ECTORY] = NONE / filename[,NEW]

,DIRSAVE = NO / YES
,UNUSED = 0 / days
,CAT[ONLY] = NO / YES
,DUP[LICATE] = YES / NO

,ATTR[IBUTES] = KEEP / RESET
,LOC[ATION] = KEEP / RESET

[,CONT[INUE] = svid / [svid,](vsn,...),...]
,TAPES / VOLUME = PO[OL] / OP[ERATOR] / vsn / (vsn,...)
,DEVICE = TAPE-C4 / device type
[,MAREN-LOCATION = <alphanum-name 1..8>]
,DRIVES = 1 / integer

,RETPD = 0 / days
,SH[ARE] = YES / NO
[,SVPASS = password]
,PRIM[ARY] = 500 / PAM pages
,SEC[ONDARY] = 100 / PAM- pages
[,BL[OCK-SIZE] = STD / MAX / PAM-Seiten]
,TARGET-ARCHIVE-VERSION / T-A-V = FROM-V26B / BEFORE-V26B [1]
,COMP[RESS] = NO / YES
,CONSISTENCY-CHECK / CONS-CHK = NO / YES

,ER[ASE] = NO / YES / ALL / ALLP

,NOW = YES / NO[,TIME = sec]
[,P[ASSWORD] = password / (password,...)]
,L[IST] = SYSLST / SYSOUT / BOTH / NONE

[1] This operand is only offered for reasons of compatibility

DIRECTORY =
This operand defines whether a directory file is used for an EXPORT run.

DIRECTORY = NONE
No directory file is used.

DIRECTORY = filename
The file with the specified name is used as a directory file.
The file must already exist and contain entries from ARCHIVE.

DIRECTORY = filename,NEW
The file with the specified name is newly created.
NEW must also be specified if the file exists but is empty, i.e. contains no entries from ARCHIVE.

DIRSAVE =
This operand determines whether the directory file is saved to the volume.

DIRSAVE = NO
The directory file is not saved.

DIRSAVE = YES
The directory file used in this run is saved as the last file of the run.

The DIRSAVE operand is only evaluated if files are found that should be saved.

UNUSED = 0 / days
This operand specifies a period of days.

If it is specified, only those files in the FILES statement which were not accessed during this period are saved. Those files which do not yet have an access date entry are not saved either.
In the case of a RESTART run, the period is calculated from the date of the RESTART run.
`days` can be a value from 0 - 36159 (99 years).

This operand is ignored for job variables.

CATONLY =
This operand specifies whether only the catalog entries of the files should be saved.

CATONLY = NO
The files and their catalog entries are saved.

CATONLY = YES
Of the files specified in the FILES statement, the catalog entries are saved only for files that are located on private volumes.
`CATONLY=YES` presupposes `DIRECTORY=NONE`, i.e. the latter is assumed even if a directory file is specified.
When job variables are saved, this operand is ignored.

Catalog entries that are saved with `CATONLY=YES` are restored with the value of the file size at the time of the save run.

DUPLICATE =
This operand defines whether files or job variables that have been specified in several FILES or JOBVAR statements should also be saved several times.

DUPLICATE = YES
Files or job variables which exist more than once are also saved more than once; a warning is issued.

DUPLICATE = NO
All duplicate files or job variables are saved only once, i.e. only the first file or job variable specified in the FILES or JOBVAR statement is processed.

`DUPLICATE=NO` is permitted only if a directory file is specified.

ATTRIBUTES =
This operand defines the password and access attributes for the saved file or job variable.

ATTRIBUTES = KEEP
The save file or job variable receives the same password and access protection attributes as the original file.

ATTRIBUTES = RESET
The password and access protection attributes are reset during the save run.
In a subsequent IMPORT run, the attributes are either reset (if no file exists in the catalog) or taken from the existing file.

LOCATION =
This operand determines how the catalog ID and user ID information for exported files should be processed.

LOCATION = KEEP
The catalog ID and user ID information of every exported file will be retained in the save file.

LOCATION = RESET
The catalog ID and user ID information is reset in the save file.

The following restrictions exist for RESET:

The EXPORT run must be executed without a directory file.

The save file created in the EXPORT run is implicitly defined as `PARAM CATID=NO` (see the PARAM statement, page 160). For this reason, `PARAM CATID=NO` must also be specified in a subsequent IMPORT or LIST statement.

A file saved in this way cannot be renamed during IMPORT.

The operand `CATONLY=YES` may not be specified.

The save file can only be duplicated with `LOCATION=RESET`.

CONTINUE =
This operand determines that an existing save version is to be extended.
The last volume of this save version is requested. The new files or job variables are written after the last file or job variable.

If the user has specified a directory file for the save run to be continued, ARCHIVE does not save those files which

- have already been saved in a later save version of the specified directory file, or
- have already been saved in the save version to be continued.

ARCHIVE includes any files which were not saved for either of these reasons in the report. This ensures that no file is saved more than once under the same svid. Any additional continuation volumes required for the output must be specified in the TAPES/VOLUME operand.

The first volume, i.e. the one to be continued, must not be specified in the TAPES/VOLUME operand. Either it must be specified in the CONTINUE operand or, if a directory file is being used, it is found automatically.

The operands DIRECTORY, TARGET-ARCHIVE-VERSION, ATTRIBUTES, LOCATION, CATONLY, SAVE-ACL, BACKUP, CHANGED, COMPRESS, CATID, SVPASS and BLOCK-SIZE must be assigned the same values as the corresponding operands of the run to be continued.
If the operands DIRSAVE, DEVICE, ERASE, LIST or SHARE do not have the same values as they had for the first run, the warning `ARC0055` is issued.

The RETPD operand is ignored in a CONTINUE run. The expiration date of the continued save version is retained.
A save version with RETPD > 0 can only be continued if the user ID has entered `TPIGNORE=YES` (JOIN).

In runs with `DIRECTORY=NONE` the user must ensure that files with the same name are not saved in the same save version, as this would make it impossible to predict which file will actually be restored in a subsequent IMPORT run.

If tapes or MT cartridges created with an ARCHIVE version < V2.8A need to be continued for some reason, they must previously be converted to the new format via tape-to-tape saving (see the FROM operand in the FILES statement, page 127).

For save versions on disk, the specification `svid,(vsn,...)` is mandatory.
The continuation disks are specified in the VOLUME operand. They must all be of the same device type. If no continuation disk is required, the VOLUME specification can be omitted. ARCHIVE aborts the run once the specified disks are full. Sufficient disks must therefore be specified in advance.

CONTINUE = svid
A directory file must have been specified. For each volume request, the last VSN is taken from the directory file.

CONTINUE = svid,(vsn,...)
If a directory file is specified for this combination of operands, ARCHIVE checks whether the specified volumes belong to this save version (svid).

– If this is not the case, ARCHIVE does not perform the save run.
– If it is the case, the last volume is requested and continued.

If no directory file is specified, ARCHIVE requests only the last volume and checks whether it belongs to the specified svid. If it does, the save version is continued.

Regardless of whether a directory file is used, it is sufficient to specify the VSN of the last volume.

CONTINUE = (vsn,...),...
If only the VSNs are specified and a directory file is being used, ARCHIVE checks whether the volumes all belong to the specified save version (svid). If this is not the case, ARCHIVE does not perform the save run.

Specifying VSNs or VSN groups does not influence the distribution of the volumes to the subtasks, the continuation volumes being automatically assigned to the subtasks in this case. They are assigned in accordance with the division of the original save run.
If no directory file is specified, no check is performed.

It is sufficient to specify the VSN of the last volume each time, regardless of whether or not a directory file is specified.

TAPES / VOLUME =
This operand determines the VSNs of the output volume.
POOL and OPERATOR are permitted only for tapes and MT cartridges.
POOL only has meaning when together with a directory file; otherwise `TAPES=OPERATOR` assumed.

TAPES / VOLUME = <u>POOL</u>
The VSNs are taken from the volume pool of the specified directory file in ascending order. If the volume pool does not contain sufficient available volumes, the operator or MAREN requests further volumes.
After deletion of the save version, the volumes are again AVAILABLE in the pool.
This operand is not permitted when saving to disk.

TAPES / VOLUME = OPERATOR
If MAREN is being used, the volume is readied by MAREN. Otherwise a VSN is requested via message `DMS0591` at the console. The specified VSN is entered in the pool with the qualifier “OPERATOR”. When the save version is deleted, the volumes are removed from the pool. This operand is not permitted when saving to disk.

TAPES / VOLUME = vsn / =(vsn,...)
The volumes are requested in the specified order.

For tapes and MT cartridges
The specified VSNs are entered in the pool with the qualifier “OPERATOR”. When the save versions are deleted, the volumes are removed from the pool. Any additionally required volumes are assigned by MAREN or by the operating.

For disks
If too few disks are specified for the save run, ARCHIVE aborts the run.

`vsn` cannot begin with `PO` or `OP` (conflict with the operand values `OPERATOR` or `POOL`).

DEVICE = TAPE-C4 / device type
This operand defines the device type for all the VSNs specified in the TAPES/VOLUME or CONTINUE operand. This also applies to VSNs which are only specified indirectly, e.g. in `CONTINUE=svid`.

MAREN-LOCATION = <alphanum-name 1..8>
This operand specifies the MAREN storage location from where free volumes should be taken if no volume is specified in the statement and no volume is available in the ARCHIVE directory pool.

If the MAREN software product is not installed or started, the operand has no effect. If the operand is specified, it takes precedence over the selection criteria as defined in the MAREN outputs.

DRIVES = 1 / integer
This defines the number of parallel runs (maximum 16). A number of output devices are used in parallel. The value of `integer` must be less than or equal to the number of devices available.

RETPD = 0 / days
This operand defines a retention period in days for the save version.
During this period
– the save version in the directory file is protected against being deleted (PURGE),
– tapes and MT cartridges are protected against being overwritten and
– the save files on disk are protected against being overwritten or deleted.

The maximum retention period is 32767 days, or about 90 years.

SHARE =
This operand defines whether the volumes or save files that have been created are given the attribute “shareable”.

SHARE = YES
The volumes or save files created are shareable.
The exported files and job variables can be read in again under a different user ID with the aid of the RENAME operand. Other users can access the volumes.

SHARE = NO
The volumes or save files created are not shareable.
The exported files and job variables can only be read in again under the user ID under which they were saved. The user cannot even read in his/her files and job variables from a save file of the system support. Other users cannot access the volumes.

SVPASS = password
This operand defines a password for the entire save version. In order to read in a file or job variable or to delete the save version (svid) via PURGE, this password must be specified either in the PASSWORD operand or in the `/ADD-PASSWORD` command.

PRIMARY = 500 / PAM pages
This operand specifies the number of blocks (2048 bytes) for the primary allocation of the save file when saving to hard disk (integer between 32 and 50331645).
When saving to tape or MT cartridge, this operand is ignored.

SECONDARY = 100 / PAM pages
This specifies the number of blocks (2048 bytes) for the secondary allocation of the save file when saving to hard disk (0 or an integer between 32 and 32767). If the value 0 is assigned and ARCHIVE tries to extend the save file, a DMS error occurs.
When saving to tape or MT cartridge, this operand is ignored.

BLOCK-SIZE =
This operand specifies the maximum number of PAM pages of the saved file that are stored in an ARCHIVE save block. `BLOCK-SIZE=16`, for example, means that up to 16 PAM pages of the saved file are written to one save block.

The operand name TAPE-BLOCK-SIZE is still supported for reasons of compatibility.

If this parameter is not specified, ARCHIVE uses the setting in the ARCHIVE parameter BLOCK-SIZE-T-C for tape cartridge devices and the setting in the ARCHIVE parameter BLOCK-SIZE-TAPE for long tapes.

BLOCK-SIZE = STD
The compatible block size of 32 kB, which corresponds to BLOCK-SIZE=15, is used.

The default value is 15 for magnetic tape or MT cartridge devices and 31 for hard disk devices. On a hard disk device, values lower than 15 will reduce the performance of ARCHIVE.

BLOCK-SIZE = MAX
The maximum block size possible in this BS2000/OSD version and for this device is used, in other words 256 kB for BS2000/OSD V6.0 or higher and TCs of type TAPE-C3 or higher, and otherwise 32 kB. When saving to public disk ARCHIVE determines the BLOCK-SIZE on the basis of the disk's maximum transfer length.

BLOCK-SIZE = PAM pages
The value range for this operand is 1...35.

– If TARGET-ARCHIVE-VERSION=BEFORE-V26B is specified, the minimum value is 1; in all other cases, it is 2.
– The maximum value is 15 for saving to tape or MT cartridge, 31 for saving to private disk, and 35 for saving to public disk.

For values greater than 31, the user must check their device configuration (with /SHOW-MASTER-CATALOG-ENTRY ...,INFORMATION=*USER) in order to ensure whether such values are permissible.

When saving NK4 disks, only odd values are processed, i.e. if an even value is specified it is decremented to the next lower odd value and a warning is issued.

The default value is 15 for magnetic tape or MT cartridge devices and 31 for hard disk devices. On a hard disk device, values lower than 15 will reduce the performance of ARCHIVE.

COMPRESS =
This operand determines whether the data should be written into the save file in a compressed form.

COMPRESS = NO
The data is not written to the save file in compressed form.

COMPRESS = YES
The data is written to the save file after being subjected to software compression.
In the case of device types which automatically implement hardware compression (e.g. TAPE-C4), COMPRESS=NO is assumed. Save files created with COMPRESS=YES cannot be read by the SIR software product.

CONSISTENCY-CHECK / CONS-CHK =
This operand determines whether check bytes are calculated for a consistency check of the save files.

CONSISTENCY-CHECK = NO
Check bytes are not calculated.

On a subsequent IMPORT or LIST operation, a `CONSISTENCY-CHECK=YES` entry is ignored.

CONSISTENCY-CHECK = YES
Check bytes are calculated and stored with the save data.
On a subsequent IMPORT or LIST operation, the correct transmission of save data is checked.

ERASE =
This operand determines whether the files and job variables are deleted after a save run.

ERASE = NO
The files and job variables are not deleted after the save run.

ERASE = YES
The exported files and job variables are deleted after the save run if they are not protected by means of a password or a retention period and if modifying access is permissible for them.

ERASE = ALL
As for YES, but those files and job variables which are protected by means of a retention period and for which only read access is permitted are also deleted.

ERASE = ALLP
This operand can only be entered with the TSOS privilege.

As for ALL, but those files and job variables which are protected by a password are also deleted, without the password having to be specified.

If both the ERASE operand and `CATONLY=YES` are specified, only the catalog entry is deleted, not the file itself.

The ERASE operand is ignored when copying save data.

File generation groups (FGGs) and file generations are deleted only if the entire FGG is being saved in this run.

NOW =
This operand specifies when the save run is to be executed.

NOW = YES
The EXPORT statement is to be executed immediately.

NOW = NO
The EXPORT statement is checked for correct syntax and then written to the queue file ARCHIVE.LATER. This process is given the ARCHIVE identifier L.yymmdd.hhmmss. The identifier is only allocated if no LATER job exists in the same task at the time. If this is not the case, then the process is stored under the existing identifier.
The LATER job can be started by means of the PROCESS statement or deleted by means of the DELETE statement.

NOW = NO,TIME=sec
The number of CPU seconds that are available to this process when it is started. This entry corresponds to the operand CPU-LIMIT in /START-(EXECUTABLE-)PROGRAM.

PASSWORD = password / (password,...)
If you specify the operand NOW=NO, you must specify passwords here for

- files or job variables to be exported which are protected by a read password,
- the save version, if it has been protected by a password with SVPASS.
- the directory file, if it is protected by a write password and is not located under TSOS.
- magnetic tapes and MT cartridges that are protected by a password (up to 4 characters) under MAREN.

In all other cases, you should enter the passwords in the /ADD-PASSWORD command before ARCHIVE is called. Only in this case are the passwords also valid for the subsequent ARCHIVE runs of the same user task.

The system support can process files and job variables without specifying the passwords. However, he must specify the password for the directory file if it is assigned to another user ID.

LIST =
This operand specifies the output medium for the report of the EXPORT run.

LIST = SYSLST
The report is output to SYSLST.

LIST = SYSOUT
In an interactive task, the report is output to the terminal.

LIST = BOTH
The report is output to both SYSLST and SYSOUT.

LIST = NONE
No report is generated.

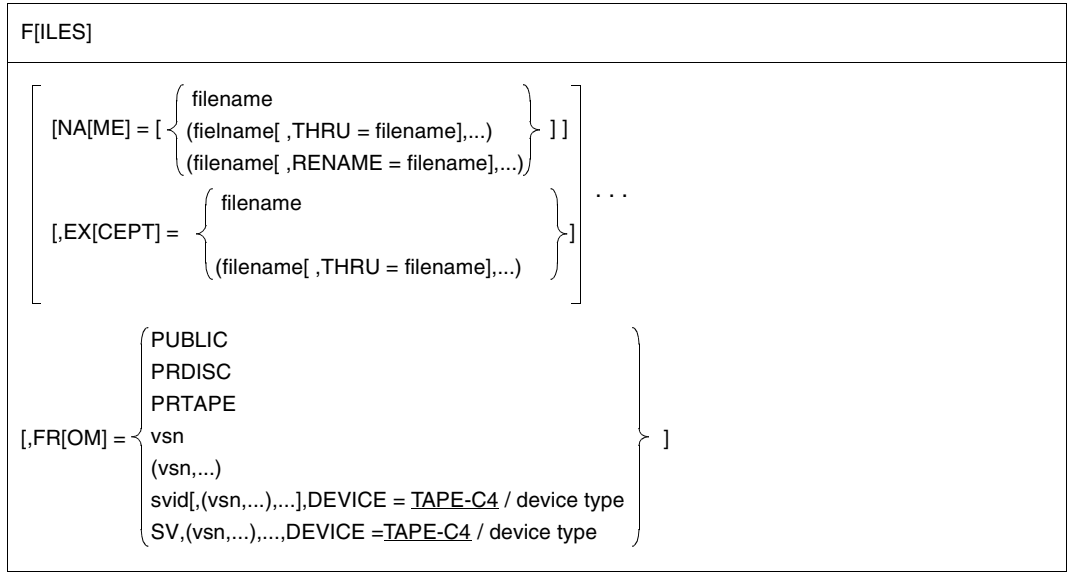
FILES
Select files in save runs

The FILES statement has two different formats, one for saving files and one for restoring files. The FILES statement for saving files is described here; the FILES statement for restoring files is described on page 130.

In the FILES statement, you must specify the files which are to be saved in an ARCHIVE run, together with the associated volumes.
More than one FILES statement can be entered. This is useful if several FROM statements are to be specified, since only one FROM entry is permitted per FILES statement.

The NAME and EXCEPT operands can be entered several times, but there may be only one EXCEPT operand for each NAME operand.

The way in which ARCHIVE processes the FILES statements and NAME operands is described in detail in the section “Parallel and serial processing” on page 75.



NAME =
The files to be saved are specified in the NAME operand. The possible specifications are dependent on the value of the CATID parameter.

- The following applies to runs with `PARAM CATID=NO`:

  No catalog ID may be entered, since the statement will otherwise be rejected. Only the catalog of the default pubset of the relevant user ID is searched.

  If the NAME operand is omitted:

  – All files which are cataloged in the default pubset of the user ID are saved.
  – If the user has the TSOS privilege, the files of all user IDs are saved from their respective default pubsets.

- The following applies to runs with `PARAM CATID=YES`:

  A catalog ID can be specified in the file name. If no catalog ID is specified, the catalog ID of the default pubset of the relevant user ID is added to the file name.
  It is also possible to specify only the catalog ID (e.g. `NAME=:c:`), in which case all files belonging to the user ID and under the specified catalog ID are saved.

  If the NAME operand is omitted:

  – All files under the relevant user ID on all available pubsets on which this ID is entered are saved.
  – If the user has the TSOS privilege, all cataloged files of all locally imported pubsets are saved. When copying save data, all the user's files are copied.
    If the user has the TSOS privilege, all saved files are copied.

  If only the catalog ID is specified, then:

  – All files of the related user ID are saved with the specified catalog ID.
    All cataloged files with the specified catalog ID are saved with the TSOS privilege.
  – When saved files are copied, all of the user's files with the specified catalog ID are copied.
    If the user has the TSOS privilege, all saved files of all user IDs with the specified catalog ID are copied.

  If, however, a fully or partially qualified file name is specified with the TSOS privilege, then:

  – Only the files with the specified catalog ID (default: default catalog ID) from the user ID TSOS are saved.

NAME = filename
If you are not working with the TSOS privilege, you can process shareable files belonging to other users with EXPORT if the operand `ATTRIBUTES=RESET` is specified.

NAME = (filename,...)
A series of file names must be enclosed in parentheses.

NAME = (filename,THRU=filename)
The specification “THRU” permits a range of files to be defined. All files with names lying alphabetically between the specified file names (both inclusive) are then addressed.
The file names must be specified in the same format (fully or partially qualified).

Notes

- All accessed files must be on the same pubset in a run with `PARAM CATID=YES`, i.e. the catalog IDs of the two files in `filename,THRU=filename` must be the same.
- If you use alias names, you must take care when using the THRU operand: the range of file names is calculated from the range output by the alias catalog.

NAME = (filename,RENAME=filename)
You can rename files with the RENAME operand. This operand is supported only for EXPORT runs without a directory (`DIRECTORY=NONE`) and for disk files.
It is not valid for catalog entries which have been saved with `CATONLY=YES` or for file generations.

For a renamed file, the entry `ERASE=YES (ALL, ALLP, ...)` is ignored with a warning message. The file is exported but not deleted.

The name in the RENAME operand replaces the original name (`NAME=...`).
If the name is partially qualified, only the part which is specified is modified.
If ARCHIVE is not running with the TSOS privilege, only files under the user's own ID can be renamed.
If the RENAME specification would result in an invalid DMS file name, it is rejected.

It is sound practice to start EXPORT runs with RENAME using `PARAM CATID=NO`; in this case, specification of a catalog ID is not permitted.

If files of several pubsets are to be exported using `PARAM CATID=YES`, the following applies:

- If, in a new file name, a catalog ID `:c:` is specified (`filename, RENAME=:c:filename`), all addressed files with the standard catalog ID of the user ID in question are written to the EXPORT volume using the catalog ID `:c:`.
- If a catalog ID `:c:` is specified in the old file name (`:c:filename, RENAME=filename`), all files concerned are written to the EXPORT volume using the same catalog ID `:c:`.

It is possible to specify user IDs instead of the file names (`$userid1., RENAME=$userid2.`).
The following applies to runs with `PARAM CATID=YES`.

- All files under the current standard catalog ID of `$userid1` are written to the EXPORT volume with `$userid2`.

ARCHIVE does not check the new catalog and user IDs.

EXCEPT = filename, / (filename,THRU=filename)
This defines files or file ranges to be excluded from the files listed in the NAME operand. An EXCEPT operand refers to the immediately preceding NAME operand.

An EXCEPT operand not preceded by a NAME operand refers to all the files under the user ID and, if the user has the TSOS privilege, to all the files in the system.

Two consecutive EXCEPT operands are not permitted. They must be separated by at least one NAME operand, which may also be empty.

If systems support want to exclude files from other user IDs, they also need to specify these user IDs.

The same rules apply to the specification of file names in the EXCEPT operand as in the NAME operand.

Examples

```
*FILES EXCEPT=HUGO1
```
This statement causes all the files under the user ID in question or, if the user has the TSOS privilege, all the files in the system, to be restored with the exception of `HUGO1`.

```
*FILES EXCEPT=(HUGO1,HUGO2,HUGO7)
```
This statement causes all the files under the user ID in question or, if the user has the TSOS privilege, all the files in the system, to be restored with the exception of `HUGO1`, `HUGO2` and `HUGO7`.

```
*FILES EXCEPT=HUGO1
*FILES EXCEPT=HUGO2
*FILES EXCEPT=HUGO7
```

This sequence of statements is not the same as the statement
`*FILES EXCEPT=(HUGO1,HUGO2,HUGO7);`
it causes all the files under the user ID(s) in question to be saved three times, namely:

– the first time, all files except the file `HUGO1`
– the second time, all files except the file `HUGO2`
– the third time, all files except the file `HUGO7`

FROM =
The FROM operand specifies the input volumes or save version from which the files are to be read.
If you do not specify FROM, all files from all pubsets and private disks which are specified in the NAME operand and all catalog entries are saved which refer to files on tapes and tape cartridges.

FROM = PUBLIC
Only those files specified in the NAME operand and residing on public volumes are saved.

FROM = PRDISC
Only those files are saved which are specified in NAME and which are on private disks.

FROM = PRTAPE
Only the catalog entries of those files are saved which are specified in the NAME operand and which are on private tapes, regardless of the CATONLY operand.
For entries of file generation groups, however, `CATONLY=YES` must be specified.
In the case of `ERASE=YES` (or `ALL...`), only the catalog entry is deleted.
RENAME is not permitted.

FROM = vsn / (vsn,...)
This determines the volumes on which the files reside.

Disks:
only those files specified in the NAME operand and having at least one extent on the specified VSNs are saved.

Tapes and MT cartridges:
specifying a tape VSN is meaningful only in runs with `CATONLY=YES` (i.e. only the catalog entries of the files residing on that tape or MT cartridge are saved).

FROM = svid[,(vsn,...),...][,DEVICE = TAPE-C4 / device type]
This form of the FROM operand is used to copy save data from ARCHIVE save runs.
`vsn,...` designates the volumes containing the ARCHIVE save runs.

If the svid to be copied was performed with the directory file specified in the DIRECTORY operand, it is sufficient to specify “svid”. If, however, no directory file is available but the svid's of the volumes are known, the operand `svid,(vsn,...)[,DEVICE=type]` is used. The DEVICE operand need only be specified when saving was to tape or MT cartridge.

The DEVICE operand may be omitted if a directory file is specified.

FROM = SV,(vsn,...),...[,DEVICE = TAPE-C4 / device type]
This form of the FROM operand is used to copy the ARCHIVE save data in whole or in part when the svid is unknown.
The DEVICE operand need only be specified if saving was to tape or MT cartridge.

Volumes specified in the VSN list must have been defined in the same save version. The VSNs must be specified in the same order as the parallel runs when no directory file is being used.

No directory file should be used when copying save data. When copying with a directory file the copy run would be entered in the directory file as the most recent save version. This might result in an unacceptably old version of a file being restored during a subsequent restoration with `FROM=LATEST`.

Only those files are copied which have been saved in their entirety to the VSNs specified.
Exception:
When the beginning of a file is on a volume with a specified VSN, the continuation tape is missing and the run was executed without a directory file.

FILES
Select files in restoration runs

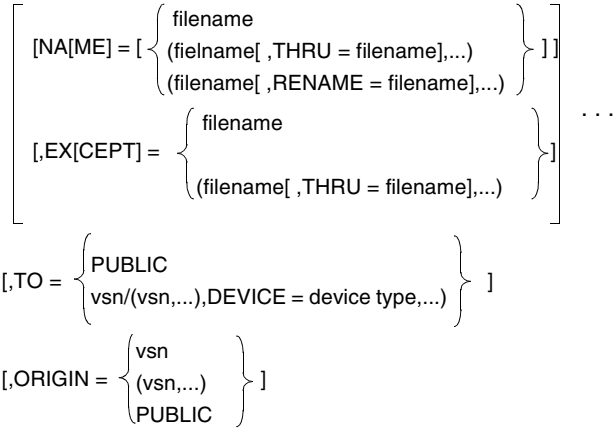
The FILES statement has two different formats, one for saving files and one for restoring files. The FILES statement for restoring files is described here; the FILES statement for saving files is described on page 124.

In the FILES statement you must specify the files which are to be restored or listed in an ARCHIVE run, together with the associated volumes.
More than one FILES statement can be entered. This is useful if several TO statements are to be specified, since only one TO entry is permitted per FILES statement.

The NAME and EXCEPT operands can be entered several times, but only one EXCEPT operand may be entered for each NAME operand.

The processing of the FILES statements and NAME operands by ARCHIVE is described in the section “Parallel and serial processing” on page 75.

F[ILES]



NAME =
The files to be read in or listed are specified in the NAME operand. The possible specifications are dependent on the value of the CATID parameter.

- The following applies to runs with `PARAM CATID=NO`:

  A catalog ID may be specified only in the RENAME operand; otherwise, the statement will be rejected.

  If the NAME operand is omitted:

  – All files are written into the default pubset of the user ID.
  – If the user has the TSOS privilege, the files of all user IDs are written to their respective default pubsets.

- The following applies to runs with `PARAM CATID=YES`:

  If no catalog ID is specified, the catalog ID of the default pubset of the relevant user ID is added to the file name.
  If a specified user ID has no entry in the JOIN file of the home pubset, this leads to an error. The files are not read in.
  It is also possible to specify only the catalog ID (e.g. `NAME=:c:`), in which case all files belonging to the user ID are restored which come from the catalog with the specified ID.

  If the NAME operand is omitted:

  – All files belonging to the appropriate user ID are restored to the pubset from which they were saved.
  – If the user has the TSOS privilege, the files of all user IDs are restored to the pubset from which they were saved.

NAME = filename
This may be a partially or fully qualified file name.

NAME = (filename,...)
When more than one file name is specified, the file names must be enclosed in parentheses.

NAME = (filename,THRU=filename)
The specification “THRU” permits a range of files to be defined. All files with names lying alphabetically between the specified file names (both inclusive) are then addressed.
The file names must be specified in the same format (fully or partially qualified).

If you use alias names, you must take care when using the THRU operand: the range of file names is calculated from the range output by the alias catalog.

In a run with `PARAM CATID=YES`, all accessed files must be on the same pubset, i.e. the catalog IDs of the two files in `filename,THRU=filename` must be the same.

NAME = (filename,RENAME=filename)
Files are renamed using the RENAME operand. The name in RENAME replaces the original name (NAME=...). If the name is partially qualified, the specified part is changed. Catalog entries saved with CATONLY=YES cannot be renamed.

If the RENAME specification would result in an invalid DMS file name, it is rejected.

If ARCHIVE is not running under the TSOS privilege, only files under the user's own ID may be renamed in RESTORE runs.
With IMPORT runs, files can also be transferred from foreign user IDs to the user's own ID and renamed, provided they have previously been saved via EXPORT.

If a catalog ID :c: is specified for the new file name (filename, RENAME=:c:filename), the following applies to runs with PARAM CATID=YES:
all referenced files with the current standard catalog ID of the user ID in question are restored to the pubset with the catalog ID :c:.

If a catalog ID :c: is specified for the old file name (:c:filename, RENAME=filename), the following applies to runs with PARAM CATID=YES:
all referenced files with the catalog ID :c: belonging to the user ID in question are restored to the current default pubset.
In the case of PARAM CATID=NO, the specification of a catalog ID (target pubset) is only possible with RENAME.

In save runs with CATONLY=YES, you can rename the catalog ID.

It is possible to specify user IDs instead of file names ($userid1., RENAME=$userid2.):

- The following applies to runs with PARAM CATID=YES:

  All files belonging to the current standard catalog ID of $userid1 are restored to the default pubset of $userid2. If you are not working with the TSOS privilege, $userid2 must be the same as the LOGON user ID.

- In the case of PARAM CATID=NO, all files of $userid1 are restored to the default pubset of $userid2.

EXCEPT = filename / (filename,THRU=filename)
This specifies the files or range of files that should be excluded from the file list specified by the NAME operand.
Only one EXCEPT operand may be specified per NAME operand.

An EXCEPT operand not preceded by a NAME operand references all the files in the user ID or, in the case of the system support, all the files in the system.
Two consecutive EXCEPT operands are not permitted. They must be separated by at least one NAME operand, which may also be empty.

The same rules apply to the specification of file names in the EXCEPT operand as in the NAME operand.

Examples

```
*FILES EXCEPT=HUGO1
```
This statement causes all the files under the user ID in question or, if the user has the TSOS privilege, all the files in the system, to be restored with the exception of HUGO1.

```
*FILES EXCEPT=(HUGO1,HUGO2,HUGO7)
```
This statement causes all the files under the user ID in question or, if the user has the TSOS privilege, all the files in the system, to be restored with the exception of HUGO1, HUGO2 and HUGO7.

```
*FILES EXCEPT=HUGO1
*FILES EXCEPT=HUGO2
*FILES EXCEPT=HUGO7
```

In contrast to the previous example, this sequence of statements causes all the files under the relevant user ID(s) to be restored three times, as follows:

– the first time, all files except the file HUGO1
– the second time, all files except the file HUGO2
– the third time, all files except the file HUGO7

In this case all the tapes have to be wound from beginning to end three times.

TO =
This specifies the output volumes.

All the volumes for the files in a FILES statement should be mounted beforehand, otherwise they will be requested during the run.

TO = (vsn,...),DEVICE=type
All files specified in the NAME operand are written to the specified VSNs.
Under DEVICE the disk type must be specified.

TO = PUBLIC
All the files, including those saved from private disks, are written to public volumes.

If the TO operand is omitted in a RESTORE run, all the files saved from **private** volumes are restored to their original volumes.

Files saved from **public** volumes are restored to public volumes.

In IMPORT runs without specification of the TO operand, the files are restored to public volumes unless otherwise specified.

If ISAM files with separate volumes for index and data are restored in a RESTORE run, the TO operand must not be specified, or at least two private disks must be specified. The index is then read to the first, the data to the second disk.

When catalog entries are being restored, the TO operand is ignored.

ORIGIN =
This operand permits you to specify that only those files are restored which were either wholly or partially on the specified volume(s) at the time of the save run.

In a restoration run with `FROM=LATEST,STATE`, ARCHIVE first requests all volumes on which all files were last saved. However, restoration only takes place from those volumes containing files to which the ORIGIN operand applies.

In the case of file generation groups (FGGs) it should be noted that the restoration of an FGG index with `REPLACE=YES` causes any existing file group with the same name to be completely deleted, but that only the file generations and the FGG index which had extents on the specified disks will be restored.
Generations on other disks which were thus lost must subsequently be restored individually.

ORIGIN = vsn / (vsn,...)
This specifies the volume serial numbers of private disks for volume-oriented restoration. Specification of public disks is not permitted.

ORIGIN = PUBLIC
This specifies public volumes for volume-oriented restoration.

HELP
Display ARCHIVE statements

With the aid of the HELP statement the user can have the syntax of all the ARCHIVE statements output to SYSOUT in German or English.

HELP
[statement[,operand] / * / keyword] [,L[ANGUAGE] = D / E]

(no entry)
A list of all ARCHIVE statements is output. Further information on these statements can be requested by means of `HELP` statements.

statement
The syntax of the specified statement is output, accompanied by a brief description of its application.

operand
Detailed information is output on the operand specified with `statement`.

*****
A list of keywords is output. Further information on these keywords can be requested by means of `HELP keyword`.

keyword
Further information on the specified `keyword` is output.

LANGUAGE =
This specifies the language in which the help texts are output.
The entry is valid until a different value is input.

LANGUAGE = D
The help texts are output in German.

LANGUAGE = E
The help texts are output in English.

By default the help texts are output in the language selected for message output. If a language is defined in the HELP statement using the LANGUAGE operand, this definition applies until a new language is defined.

Example

```
HELP *,LANGUAGE=E
-> ANWEISUNGEN,
-> ARBEITSDATEIEN,
-> ARCHIVE,
-> ARCHIVE-DATEIEN,
-> ARCHIVE,CATID
-> ARCHIVE,CNS
-> ARCHIVE,DEST
-> ARCHIVE,OP
-> ARCHIVE,RES
-> ARCHIVE,SNR
-> ARCHIVE,STREAM
-> ARCHIVE,UNL
-> ARCHIVE,WRCHK
-> AUSGABE,
-> BANDRESERVIERUNG,
-> BEISPIEL,
-> BEISPIEL,MACRO
-> CHKPT-DATEI,
-> CONTINUATION CHARACTER,
MORE ? (Y/N) :
N


HELP STATEMENTS
-> STATEMENTS,
THE FOLLOWING  A R C H I V E  COMMANDS ARE AVAILABLE:
 ------------------------------------------------------------------
  SAVE         RESTORE      DIRECTORY    PROCESS      OTHERS
                            ADMINISTR.   ADMINISTR.
  ------------ ------------ ------------ ------------ ------------
  FILES        FILES        INQUIRE      STATUS       PARAM
  JOBVAR       JOBVAR       POOL         PROCESS      SHOW-DEFAULT
  SAVE         RESTORE      PURGE        DELETE       HELP
  EXPORT       IMPORT                                 END
               LIST
 ------------------------------------------------------------------
MORE ? (Y/N) :
Y
-------   survey of ARCHIVE statements   -----------------------------------
DELETE       : erase waiting ARCHIVE processes from ARCHIVE.LATER file
END          : terminate program ARCHIVE
EXPORT       : save for data transfer to an other user environment
FILES        : select files for save or restore
HELP         : display help texts to explain the use of ARCHIVE
IMPORT       : restore save from an other user environment
```

```
INQUIRE      : show information out of an directory file
JOBVAR       : select jobvariables for save or restore
LIST         : show the contents of save files
PARAM        : set operands for further ARCHIVE processing
POOL         : administrate volume pool of directory file
PROCESS      : start ARCHIVE processes from ARCHIVE.LATER file
PURGE        : erase save versions from directory file
RESTORE      : restore from a save file
SAVE         : save data to tape, MBK or disc
STATUS       : show status of ARCHIVE processes
SHOW-DEFAULT : show default values of ARCHIVE parameters in par. file
```

```
HELP INQUIRE
-> INQUIRE,
THE "INQUIRE" COMMAND ENABLES THE USER TO LIST
INFORMATION CONTAINED IN A DIRECTORY.
------------------------------------------------------------------------------
statement   operands
------------------------------------------------------------------------------
INQUIRE / Q
            < DIR<ECTORY> = !$TSOS.ARCHIVE.DIR! / filename>

            < <,F<ILES>  = filename / (filename,...) > /
              <,J<OBVAR> = jv-name / (jv-name,...) > /
              <,SV       = ALL / OBS<OLETE> / svid / (svid,...) /
                           backup,date<.time>
                              <,F<ILES>><,J<OBVAR>>

              <,POOL <,AV<AILABLE>> > >

            <,P<ASSWORD>  = password>
            <,L<IST>      = !SYSOUT! / SYSLST / BOTH >
------------------------------------------------------------------------------
```

IMPORT
Transfer saved data

The IMPORT statement permits files, catalog entries or job variables which were saved to tape, disk or MT cartridge in the same EXPORT run to be transferred to another BS2000/OSD system or to another user ID (RENAME).

The files and/or job variables to be processed must be specified with the FILES and/or JOBVAR statement before the IMPORT statement is entered.

I[MPORT]
DIR[ECTORY] = NONE / filename ,FR[OM] = svid / [svid,](vsn,...),... ,DEVICE = TAPE-C4 / device type ,DRIVES = 1 / integer ,CONSISTENCY-CHECK / CONS-CHK = NO / YES ,REP[LACE] = NO / YES / ALL / ALLP ,SPACE = REORG / KEEP ,CONV[ERSION] = STD / NO / CONV ,NOW = YES / NO[,TIME = sec] [,P[ASSWORD] = password / (password,...)] ,L[IST] = { SYSLST / SYSOUT / BOTH / NONE } [,ALL]

DIRECTORY =
This specifies whether a directory file is used for an IMPORT run.

DIRECTORY = NONE
Unless otherwise specified, IMPORT runs are executed without a directory file.

DIRECTORY = filename
Name of the directory file used for the EXPORT run. The file must first be imported in a separate IMPORT run with `DIRECTORY=NONE`.

FROM =
This specifies the input volumes or save version from which the files or job variables are to be read. The FROM operand must be specified for IMPORT runs.
Only those files are read in which are stored in full on the specified volumes.

When importing a volume created by HSMS in `several-svid` mode, the SVID (not the SFID) must be specified.

FROM = svid
Every file or job variable requested in the FILES or JOBVAR statement and belonging to this save version (svid) is read in. A directory file must be specified.

FROM = (vsn,...)
Every required file or job variable on the specified volumes is imported.
The volumes must be listed in the order in which they were recorded during export.
If a directory file is used, only those files located in their entirety on the specified volumes are read in.

FROM = (vsn,...),(vsn,...)
When files or job variables saved in parallel runs are imported, the volumes must be grouped according to the parallel runs, i.e. the volumes of a parallel run must be specified in each parentheses in the order in which they were written to.

Data of an export run which was saved using

```
FILES
EXPORT ...,DRIVES=2,TAPES=(vsn1,vsn2)
```

and distributed according to the EXPORT listing in two parallel runs over `vsn1` and `vsn2` must be transferred as follows:

```
FILES
IMPORT ...,DRIVES=2,FROM=(vsn1),(vsn2)
```

An attempt to import the files with

```
IMPORT ...,DRIVES=2,FROM=(vsn1,vsn2)
```

causes the run to abort because ARCHIVE cannot allocate the volumes correctly.

FROM = svid,(vsn,...)
See previous operand value.
In addition, if a directory file is specified, ARCHIVE checks whether the specified volumes belong to this save version. Should this not be the case, the run is aborted.
This entry is mandatory for IMPORT runs from disk.

DEVICE = TAPE-C4 / device type
This operand is used to define the device type. It must be the same as the one used for the EXPORT run.
The DEVICE operand defines the device type for all the VSNs. This also applies to VSNs only specified indirectly, e.g. in `FROM=svid`.

DRIVES = 1 / integer
This specifies the number of parallel runs (maximum 16).
The number of parallel runs should correspond to the number defined for the appropriate EXPORT run.

CONSISTENCY-CHECK / CONS-CHK =
This determines whether a consistency check of the save data is carried out.

CONSISTENCY-CHECK = NO
No consistency check is carried out.

CONSISTENCY-CHECK = YES
Before transferring save data, the check bytes are calculated and a comparison is made with the check bytes stored with the save data during EXPORT. If an error occurs, the message `ARC0413` is output.

This operand value is ignored if the EXPORT operation was carried out with the `CONSISTENCY-CHECK=NO` operand.

REPLACE =
This specifies whether or not existing job variables and files are to be replaced.

REPLACE = NO
A file or job variable is not read in if one with the same name already exists under this user ID. In this case a message is output.

REPLACE = YES
If a file or job variable with the same name already exists, it is replaced by the file or job variable from the save run.

However, a file or job variable is not replaced if
- only read access is permitted for it,
- it has an expiration date (EXPIR DATE) in the catalog which is later than the current date, or
- it is protected by a password which has not been specified.

REPLACE = ALL
If a file or job variable with the same name already exists, it is replaced by the saved file or job variable. The file or job variable is replaced even if only read access is permitted for it or if the expiration date has not yet been reached. If the existing file is protected by a password, however, the password must be specified.

REPLACE = ALLP
This operand is only permitted with the TSOS privilege. It permits files or job variables protected by passwords to be replaced by the corresponding files from the save run without the passwords having previously been specified.
Otherwise the same applies to the ALLP operand as for ALL.

SPACE =
This specifies whether or not files to be replaced by files with the same names from the save run are to be deleted before restoration.

SPACE = REORG
Files to be replaced by files with the same names from the save run are deleted before restoration. Usually the file then occupies a different disk extent after restoration. Thus restoration runs with `SPACE=REORG` have the effect of reorganizing the disks.

SPACE = KEEP
Files to be replaced by files with the same names from the save run are overwritten in the extents which they occupy at the time of the restoration.
The specification `SPACE=KEEP` overrides any definitions of storage areas for the file to be written back.

Example

The file `HUGO` is to be restored with `SPACE=KEEP`. Let us assume that there is already a file of this name on the private disk `PRIV01`. If `TO=PUBLIC` is then specified for this run in the FILES statement, ARCHIVE still writes the file back to the private disk PRIV01. In other words the specification `SPACE=KEEP` overrides the specification `TO=PUBLIC`.

The specifications in the SPACE operand are ignored if `REPLACE=NO` has also been specified.

Specifications in the SPACE operand are ignored for file generation groups. FGGs are always deleted before restoration.

The following points must be borne in mind for ISAM files with a separate file index section and file data section:
- The file in the system must have the same logical structure as the saved file (i.e. the index blocks and data blocks must occupy the same logical blocks).
- The access method must be ISAM.

The SPACE operand is irrelevant for catalog entries for files on tape or MT cartridge. The catalog entries are always deleted when `REPLACE=YES` is specified.

CONVERSION =
This operand controls whether files which had a PAM key when they were saved must be converted if the output volume is an NK disk (see the table on page 97).
The operand is ignored for job variables.
Entering `CONVERSION=STD / CONV` requires the PAMINT subsystem.
In the case of partially saved files, only the CONV entry is possible.
If a conversion file which is to be imported already exists, it is always reorganized; the SPACE operand is ignored.

CONVERSION = STD
In IMPORT runs to an NK disk, files with a PAM key are converted by means of PAMINT in accordance with the following rules:

- K-ISAM files to NK-ISAM files
  (BLOCK-CONTROL-INFO=WITHIN-DATA-BLOCK)
- K-SAM files to NK-SAM files
  (BLOCK-CONTROL-INFO=WITHIN-DATA-BLOCK)
- K-UPAM files to NK-UPAM files
  (BLOCK-CONTROL-INFO=NO)
  PAM files whose structure is known (load modules, libraries) are converted accordingly; in the case of PAM files whose structure is not known, the PAM key information is lost. If the PAM key contained information, an appropriate message is issued.

CONVERSION = NO
Files with a PAM key are not converted in IMPORT runs to an NK disk and are thus not restored.

CONVERSION = CONV
In an IMPORT run to an NK disk, files with PAM keys are converted to “CONV format” by means of PAMINT. The restored file contains all the PAM keys at the end of the file in separate blocks.

NOW =
This operand specifies when the IMPORT statement is to be executed.

NOW = YES
The IMPORT statement is executed immediately.

NOW = NO
The IMPORT statement is checked for syntax errors and the written to the queue file ARCHIVE.LATER. This process is given the ARCHIVE identifier L.yymmdd.hhmmss. The identifier is only allocated if no LATER job exists in the same task at the time. If this is not the case, then the process is stored under the existing identifier.
The LATER job can be started by means of a PROCESS statement or deleted by means of a DELETE statement.

NOW = NO,TIME=sec
The number of CPU seconds that are available to this process when it is started. This entry corresponds to the operand CPU-LIMIT in /START-(EXECUTABLE-)PROGRAM.

PASSWORD = password / (password,...)
If you specify the operand NOW=NO, you must enter passwords here for

- files or job variables to be imported which are protected by read passwords;
- the directory file, if it is protected by a read password and is not under TSOS;
- the save version, if it has been protected by a password with SVPASS.
- magnetic tapes and MT cartridges that are protected by a password (max. 4 characters) under MAREN.

In all other cases, you should enter the passwords with the /ADD-PASSWORD command before ARCHIVE is called. Only in this case are the passwords also valid for the subsequent ARCHIVE runs of the same user task.

The system support can process files and job variables without specifying the passwords. The password for the directory file must, however, be specified if it is assigned to a foreign user ID.

LIST =
This defines the output medium for the report of this ARCHIVE run.

LIST = SYSLST
The report is output to SYSLST.

LIST = SYSOUT
In an interactive task, the report is output to the terminal.

LIST = BOTH
The report is output to both SYSLST and SYSOUT.

LIST = NONE
No report is generated.

LIST = ...,ALL
All the file names and job variables on the volumes read in are listed. If a save version (svid) has also been specified in the FROM operand, all the files belonging to this save version are listed (not only those specified in the FILES and/or JOBVAR statements).

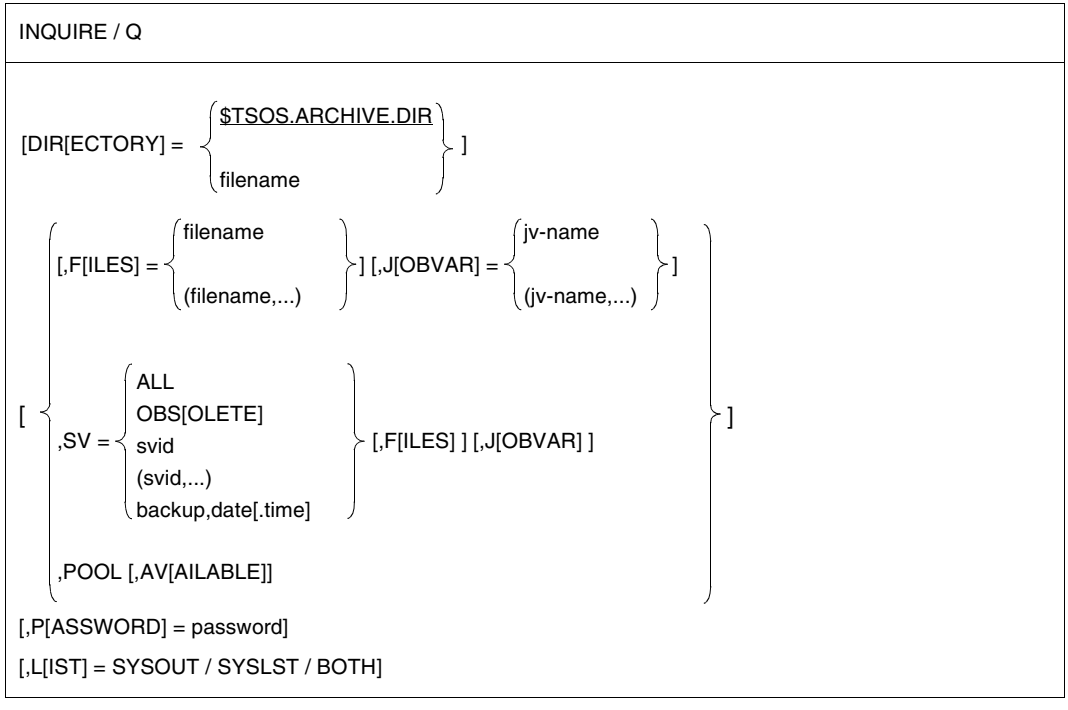
IMPORT statement notes

- After an EXPORT run to disk, the save file ARCHIVE.SAVE.FILE.date.time.vsn is deleted from the catalog. It must therefore be imported via the IMPORT-FILE command before an IMPORT run.
- Catalog entries which were saved in an EXPORT run with CATONLY=YES are not restored if RENAME=... is specified for them in the FILES statement when reading the files in.
- If the FILES statement does not contain a TO operand, the files are written to public volumes, irrespective of which volume the files resided on before the save operation.

INQUIRE
Display contents of directory file

The INQUIRE statement enables the user to output the contents of the directory file.

If neither the FILES, JOBVAR, SV nor POOL operand is specified, all files and then all job variables are output in alphabetical order.
If FILES and JOBVAR or the SV operand are specified, files and job variables are likewise output in merged form.
Files and job variables cannot be output separately unless separate INQUIRE statements with the appropriate operands are entered.



DIRECTORY= $TSOS.ARCHIVE.DIR / filename
This is the name of the directory file whose contents are to be output.
The user can address any directory file created under his own user ID. He can also address his own files in the system support's directory file.

FILES = filename / (filename,...)
This determines that information about files is output from the directory file.
The fully or partially qualified names of the desired files should be entered for `filename`.
If the user has specified the directory file of the system support, only information about the files under his own user ID is supplied.

JOBVAR = jv-name / (jv-name,...)
This determines that information relating to job variables is output from the directory file. The fully or partially qualified names of the job variables should be entered for `jv-name`.

SV =
This determines that information relating to save versions (svid) is output from the directory file.

SV = ALL
All the save versions (svid) in the specified directory file are output.

SV = OBSOLETE
All the “obsolete” save versions (svid) are output from the directory file. An obsolete save version is one whose retention period (RETPD) has expired.

SV = svid / (svid,...)
Only the specified save versions are output from the directory file.

SV = backup,date.time
All the save versions created by means of the SAVE statement and with this BACKUP entry up to and including the specified time are output.

SV = ...,FILES
The file names are output sorted according to save version. The user only receives information about his own files. The system support on the other hand is given information about all the files in the system.

SV = ...,JOBVAR
The job variable names are output, sorted according to save version. Any user other than the system support receives a list of his own job variables.

POOL
This outputs the VSNs of the volumes in the volume pool in the directory file in ascending order. The following information is also output:

– volume type
– svid, if the volume has already been included in a save version
– AVAILABLE, if the volume is free
– UNUSABLE, if the volume cannot be used for writing
– POOL, if the volume was added to the pool by means of the POOL statement
– OPERATOR, if the volume was added to the pool by the operating
– the expiration date, if it has not yet been reached

POOL,AVAILABLE
Only the VSNs of the available volumes are output.

PASSWORD = password
The password for the ARCHIVE run.
An entry is necessary if the directory file is protected by a read password and is not under TSOS.

LIST =
This defines the output medium for the report of this ARCHIVE run.

LIST = SYSOUT
The report is output to the user's terminal in interactive mode.
This is the default for runs in interactive mode.

LIST = SYSLST
The report is output to SYSLST.
This is the default for batch mode.

LIST = BOTH
The report is output to both SYSLST and SYSOUT.

Example 1

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*INQUIRE DIRECTORY=ARCHIVE.DIR,FILES=FILE. ------------------------------  (1)
*   I N Q U I R E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES

INQUIRE DIRECTORY=ARCHIVE.DIR,FILES=FILE.
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100814.150638',
           VERSION='9.0A'
  ***     I N Q U I R E   A R C H I V E   R E P O R T  ***      10-08-14     PAGE    1 ---  (2)
                          DIRECTORY - $TSOS.ARCHIVE.DIR
                          ***   USER - TSOS       ***
        FILE/JOB VARIABLE NAME          VERS    CFID     SAVE VERSION    EXP.DATE TYPE  VSN(S)

FILE.1                                     1  C1C6D955  S.100810.140853   (OBS)   FULL  SBZ3.2
FILE.2                                     1  C1C6D956  S.100810.140853   (OBS)   FULL  SBZ3.1
FILE.3                                     1  C1C6D957  S.100810.140853   (OBS)   FULL  SBZ3.2
  ***     E N D   O F   A R C H I V E   R E P O R T    ***      10-08-14     PAGE    2
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) ARCHIVE should output all information relating to the `FILE.` files from the directory file `ARCHIVE.DIR`.

(2) The ARCHIVE report is output to SYSOUT.
It contains the following information:

- FILE/JOB VARIABLE NAME
  All the files under the user ID `TSOS` having the partially qualified file name `FILE.` and included in the directory file `ARCHIVE.DIR`.
- CFID: Internal file names.
- SAVE VERSION: Name of the save versions.
- EXP.DATE:
  expiration date; the retention period of the volume has expired (OBS).
- TYPE: Save types.
- VSN(S): VSNs of the input volumes.

Example 2

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*INQUIRE DIRECTORY=ARCHIVE.DIR,POOL ------------------------------------  (1)
*   I N Q U I R E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES

INQUIRE DIRECTORY=ARCHIVE.DIR,POOL
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100814.150710',
           VERSION='9.0A'
  ***    I N Q U I R E   A R C H I V E   R E P O R T  ***     10-08-14    PAGE    1 -----  (2)
                          DIRECTORY - $TSOS.ARCHIVE.DIR
 VSN      VOL.TYPE        CURRENT USE        OWNER         EXP.DATE      REMARK

OS0320    TAPE-C4       S.100810.140853      POOL
OS0321    TAPE-C4         AVAILABLE          POOL
      ***     E N D   O F   A R C H I V E   R E P O R T    ***      10-08-14      PAGE    2
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) ARCHIVE should output information relating to the volume pool from the directory file ARCHIVE.DIR.

(2) The ARCHIVE report is output via SYSOUT. It contains the following information:

- VSN: VSNs of the save volumes contained in the volume pool in the directory file ARCHIVE.DIR.
- VOL.TYPE: device type of the respective volume.
- CURRENT USE:
  The corresponding save version is listed for MT cartridge OS0320. MT cartridge OS0321 is still available (AVAILABLE).
- OWNER:
  The MT cartridges are entered in the volume pool with the POOL statement.
- EXP.DATE
  The expiration date has already been reached, so no entry is shown. Otherwise the expiration date is shown here.

Example 3

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*INQUIRE DIRECTORY=ARCHIVE.DIR,SV=OBSOLETE ------------------------------  (1)
*   I N Q U I R E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES

INQUIRE DIRECTORY=ARCHIVE.DIR,SV=OBSOLETE
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100814.150820',
           VERSION='9.0A'
  ***    I N Q U I R E   A R C H I V E   R E P O R T  ***     10-08-14     PAGE    1 --  (2)
                              DIRECTORY - $TSOS.ARCHIVE.DIR
SAVE VERSION IDENTIFIER - S.100810.140853 - ON VOLUME TYPE TAPE-C4  (OBSOLETE)

OPTIONS:- CATONLY=NO ,ERASE=NO ,CHANGED=YES
          SHARE=NO ,BACKUP=D,RETPD=    0,DRIVES=  1
S/SAVE  VSN(S)
   0    OS0320
   ***     E N D   O F   A R C H I V E   R E P O R T   ***     10-08-14     PAGE    2
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) ARCHIVE is to output all the information on the obsolete save versions from the directory file ARCHIVE.DIR.

(2) The report on the ARCHIVE run is output via SYSOUT and contains the following information:

- SAVE VERSION IDENTIFIER - S... - ON VOLUME TYPE TAPE-C4 (OBSOLETE)
  The specified save version in the directory file ARCHIVE.DIR is released for deletion; its retention period has expired.
- OPTIONS:
  Operands which were valid in the SAVE statement for the expired save version. Only YES or NO is displayed here for the ERASE operand. YES is also displayed if ERASE=ALL/ALLP/ALLPCNS/ALLCNS was specified in the SAVE statement.
- S/SAVE: The number of the parallel run.
- VSN(S): VSNs of the volumes of this save version.

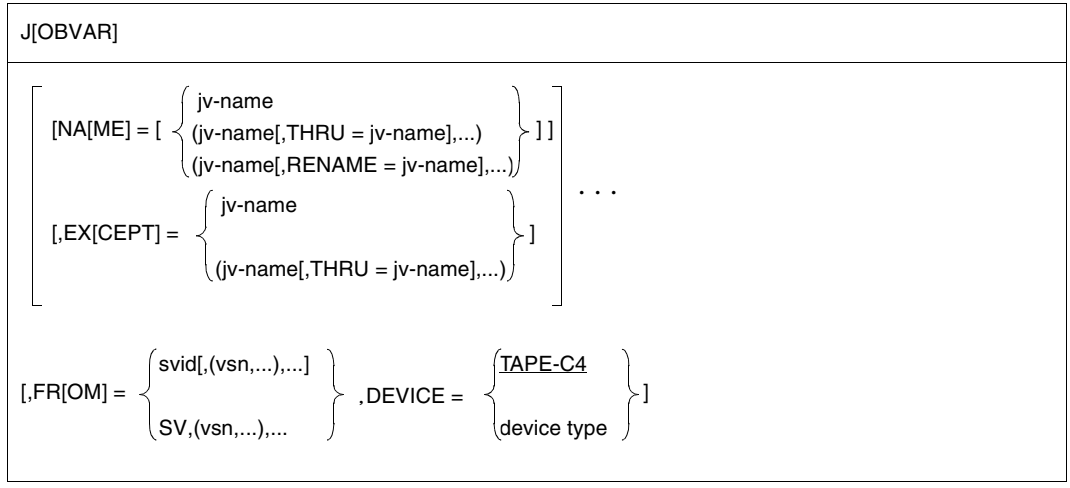
JOBVAR
Select job variables in save runs

The JOBVAR statement has two different formats, one for saving job variables and one for restoring job variables. The JOBVAR statement for saving job variables is described here; the JOBVAR statement for restoring job variables is described on page 154.

In the JOBVAR statement, you must specify the job variables to be saved in an ARCHIVE run, together with the associated volumes.

More than one JOBVAR statement can be entered. Each JOBVAR statement may contain a number of NAME and EXCEPT operands, but only one EXCEPT operand may be specified per NAME operand.

The handling of the JOBVAR statement and the NAME operand is analogous to that described for the FILES statement in the section “Parallel and serial processing” on page 75.



NAME =
The job variables to be saved are specified in the NAME operand. The possible specifications are dependent on the value of the CATID parameter.

- The following applies to runs with `PARAM CATID=NO`:

  No catalog ID may be entered, otherwise the statement will be rejected. Only the catalog of the default pubset of the relevant user ID is searched.

  If the NAME operand is omitted:

  – All job variables which are cataloged in the default pubset of the user ID are saved.
  – If the user has the TSOS privilege, the job variables of all user IDs are saved from their respective default pubsets.

- The following applies to runs with `PARAM CATID=YES`:

  A catalog ID can be specified in the job variable name. If no catalog ID is specified, the catalog ID of the default pubset of the relevant user ID is added to the job variable name. It is also possible to specify only the catalog ID (e.g. `NAME=:c:`), in which case all job variables belonging to the user ID and under the specified catalog ID are saved.

  If the NAME operand is omitted:

  – All job variables under the relevant user ID on all available pubsets on which this ID is entered are saved.
  – All cataloged job variables of all locally imported pubsets are saved if the user has the TSOS privilege.
  – When copying save data, all job variables of the user are copied.
    All saved job variables are copied if the user has the TSOS privilege.

NAME = jv-name
This may be a partially or fully qualified job variable name.

NAME = (jv-name,...)
If more than one job variable is specified, the names must be enclosed in parentheses.

NAME = (jv-name,THRU=jv-name)
The specification “THRU” enables a range of job variables to be defined. All those lying alphabetically between the job variable names (both inclusive) are then addressed.

The job variables must be on the same pubset for a run with `PARAM CATID=YES`, i.e. the catalog IDs of the two job variables in `jv-name,THRU=jv-name` must be the same.

NAME = (jv-name,RENAME=jv-name)
You can rename job variables with the RENAME operand. This operand is supported only for EXPORT runs without a directory file.
For a renamed job variable, the entry `ERASE=YES (ALL, ALLP, ...)` is ignored with a warning message. The job variable is exported but not erased.

The name in the RENAME operand replaces the original name (`NAME=...`).
If the name is partially qualified, the part which is specified is modified.
If ARCHIVE is not running under the TSOS privilege, only job variables under the user's own ID are renamed.

It is sound practice to start EXPORT runs using RENAME with `PARAM CATID=NO`; in this case, specification of a catalog ID is not permitted.

If job variables of several pubsets are to be exported using `PARAM CATID=YES`, the following applies:

- If, in a new job variable name, a catalog ID `:c:` is specified (`jv-name,RENAME=:c:jv-name`), all addressed job variables with the standard catalog ID of the user ID in question are written to the EXPORT volume using the catalog ID `:c:`.
- If a catalog ID `:c:` is specified in the old job variable name (`:c:jv-name,RENAME=jv-name`), all job variables concerned are written to the EXPORT volume using the same catalog ID `:c:`.

It is possible to specify user IDs instead of the job variable names (`$userid1.,RENAME=$userid2.`).
The following applies to runs with `PARAM CATID=YES`: All job variables under the current standard catalog ID of `$userid1` are written to the EXPORT volume with `$userid2`. ARCHIVE does not check the new catalog and user IDs.

EXCEPT = jv-name / (jv-name,THRU=jv-name)
This specifies job variables or job variable ranges that are to be excluded from the list of job variables in the NAME operand. Only one EXCEPT operand may be specified per NAME operand.

If the EXCEPT operand is not preceded by a NAME operand, it refers to all the job variables under the user ID or, in the case of the system administrator ID, to all the job variables in the system.

The same applies to the EXCEPT operand in the JOBVAR statement as was described for the EXCEPT operand in the FILES statement.

The rules governing the specification of job variable names in the NAME operand also apply to the EXCEPT operand.

FROM =
This operand must and can only be specified when job variables are to be copied from an existing save version to a new save version. Only the job variables which are specified in the NAME operand and fulfill the conditions of the FROM operand are copied.

FROM = svid
This specifies the save version in which the job variables in the NAME operand were saved. Specification of a directory file is mandatory.

FROM = svid,(vsn,...)
Only the job variables in the NAME operand which belong to a save version on the volume with the specified VSN are copied. If a specified VSN does not belong to the specified save version, the ARCHIVE run is aborted.
A directory file must not be specified.

FROM = SV,(vsn,...)
This specification is used to duplicate save data fully or partially when the svid is not known. All the job variables given in the NAME operand and resident on the specified volumes (`vsn,...`) are copied. This operand is not permitted when saving to disk.

FROM = ...,DEVICE = TAPE-C4 / device type
If the svid is not known, the device type of the save volume is specified (not possible when saving to disk).

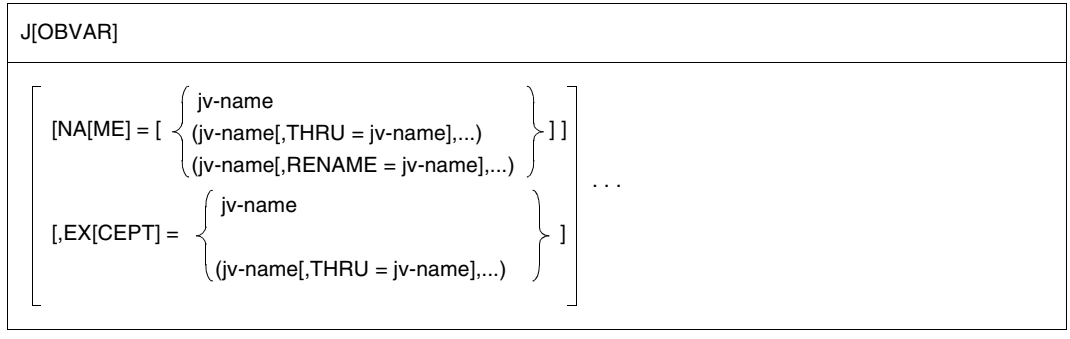
JOBVAR
Select job variables in restoration runs

The JOBVAR statement has two different formats, one for saving job variables and one for restoring job variables. The JOBVAR statement for restoring job variables is described here; the JOBVAR statement for saving job variables is described on page 150.

In the JOBVAR statement the job variables to be read in in an ARCHIVE run are specified.

More than one JOBVAR statement can be specified. Each JOBVAR statement may contain a number of NAME and EXCEPT operands, but only one EXCEPT operand per NAME operand.

The handling of the JOBVAR statement and the NAME operand is analogous to what was described for the FILES statement in the section “Parallel and serial processing” on page 75.



NAME =
The job variables to be restored are specified in the NAME operand.
The possible specifications are dependent on the value of the CATID parameter.

- The following applies to runs with `PARAM CATID=NO`:

  A catalog ID may be specified only in the RENAME operand. Otherwise, the statement will be rejected.

  If the NAME operand is omitted:

  – All job variables are written into the default pubset of the user ID concerned.
  – If the user has the TSOS privilege, the job variables of all user IDs are written into their respective default pubsets.

- The following applies to runs with `PARAM CATID=YES`:

  If no catalog ID is specified, the catalog ID of the default pubset of the relevant user ID is added to the job variable name.
  If a specified user ID has no entry on the home pubset, this leads to an error. The job variables are not read in.
  It is also possible to specify only the catalog ID (e.g. `NAME=:c:`), in which case all job variables belonging to the user ID are restored which come from the catalog with the specified ID.

  If the NAME operand is omitted:

  – All job variables under the relevant user ID are restored to the pubset from which they were saved.
  – If the user has the TSOS privilege, the job variables of all user IDs are restored to the pubset from which they were saved.

NAME = jv-name
This may be a partially or fully qualified job variable name.

NAME = (jv-name,...)
If more than one job variable name is specified, they must be enclosed in parentheses.

NAME = (jv-name,THRU=jv-name)
The specification “THRU” enables a range of job variables to be defined. All the job variables lying alphabetically between the two names (both inclusive) are then addressed.

The job variables must be on one pubset in a run with `PARAM CATID=YES`, i.e. the catalog IDs of the two job variables in `jv-name,THRU=jv-name` must be the same.

NAME = (jv-name,RENAME=jv-name)
Using the RENAME operand, job variables can be renamed when they are read in in RESTORE and IMPORT runs. The name specified in RENAME replaces the original name (NAME=...). If the name given is partially qualified, only the specified part is renamed. If no catalog ID :c: is specified in the job variable names, the standard catalog ID is assumed.

A catalog ID (target pubset) may only be specified in RENAME in the case of PARAM CATID=NO.

EXCEPT=jv-name / (jv-name,THRU=jv-name)
This specifies job variables or job variable ranges that are to be excluded from the list of job variables in the NAME operand.
Only one EXCEPT operand may be specified per NAME operand.

If the EXCEPT operand is not preceded by a NAME operand, it refers to all the job variables under the user ID or, in the case of the system administrator ID, to all the job variables in the system.

The same applies to use of the EXCEPT operand in the JOBVAR statement as was described for the EXCEPT operand in the FILES statement.

The rules governing the specification of job variable names in the NAME operand also apply to the EXCEPT operand.

LIST
List the contents of save volumes and save versions

The LIST statement is used to list files, job variables and catalog entries saved on one or more volumes of one save run.

You must specify the files and/or job variables to be processed with FILES and/or JOBVAR statements before you enter the LIST statement.

```
L[IST]

DIR[ECTORY] = NONE / filename

,FR[OM] = svid / [svid,](vsn,...),...
[,DEVICE = device type]
,DRIVES = 1 / integer

,CONSISTENCY-CHECK / CONS-CHK = NO / YES

,NOW = YES / NO[,TIME = sec]
[,P[ASSWORD] = password / (password,...)]
,L[IST] = { SYSLST / SYSOUT / BOTH / NONE } [,ALL]
```

DIRECTORY =
This defines whether a directory file is used for the ARCHIVE run.

DIRECTORY = NONE
Unless otherwise specified, LIST runs are performed without a directory file.

DIRECTORY = filename
The name of the directory file used for the SAVE or EXPORT run is specified here.

FROM =
This specifies the input volumes or save version from which the files or job variables are to be read.

FROM = svid
Every file or job variable requested by a FILES or JOBVAR statement and belonging to this save version (svid) is listed. The specification of a directory file is mandatory.

FROM = (vsn,...),...
`vsn` is the volume serial number of the volume from which the files in the FILES statement and job variables in the JOBVAR statement are to be listed. This operand is not permitted when saving to disk.
The VSNs must be grouped according to the parallel runs, i.e. the VSNs specified (in the right order) in parentheses must belong to the same subsave run.

FROM = svid,(vsn,...),...
As above.

In addition, if a directory file is specified, ARCHIVE checks whether the volumes specified belong to this save version. If not, the run is aborted.

DEVICE = device type
This operand specifies the device type. This must be the same type as was specified for the save run.
The DEVICE operand defines the device type for all the VSNs, including those specified only indirectly, e.g. in `FROM=svid`.
The default value is `TAPE-C4` with `DIRECTORY=NONE`; otherwise the device type is taken from the directory.

DRIVES = 1 / integer
This specifies the number of parallel runs (up to 16).

CONSISTENCY-CHECK / CONS-CHK =
This determines whether a consistency check of the save data is carried out.

CONSISTENCY-CHECK = NO
No consistency check is carried out.

CONSISTENCY-CHECK = YES
Before transferring save data, the check bytes are calculated and a comparison is made with the check bytes stored with the save data during EXPORT. If an error occurs, the message `ARC0413` is output.

This operand value is ignored if the EXPORT or SAVE operations were carried out with the `CONSISTENCY-CHECK=NO` operand.

NOW =
This operand specifies when the LIST statement is to be executed.

NOW = YES
The LIST statement is executed immediately.

NOW = NO
The LIST statement is checked for correct syntax and then written to the queue file `ARCHIVE.LATER`. This process is given the ARCHIVE identifier `L.yymmdd.hhmmss`.
The identifier is only allocated if no LATER job exists in the same task at the time. If this is not the case, then the process is stored under the existing identifier.
The LATER job can be started by means of a PROCESS statement or deleted by means of a DELETE statement.

NOW = NO,TIME=sec
The number of CPU seconds that are available to this process when it is started. This entry corresponds to the operand CPU-LIMIT in /START-(EXECUTABLE-)PROGRAM.

PASSWORD = password / (password,...)
If you specify the operand NOW=NO, you must enter passwords here for

- the directory file, if it is protected by a write password and is not under TSOS.
- the save version, if it has been protected by a password with SVPASS.
- magnetic tapes and MT cartridges that are protected by a password (up to 4 characters) under MAREN.

In all other cases, you should enter the passwords with the /ADD-PASSWORD command before ARCHIVE is called. Only in this case are the passwords also valid for the subsequent ARCHIVE runs of the same user job.

The system support must specify the password for the directory file if the file is assigned under another user ID.

LIST =
This defines the output medium for the report of this ARCHIVE run

LIST = SYSLST
The report is output to SYSLST.

LIST = SYSOUT
In interactive mode, the report is output to the terminal.

LIST = BOTH
The report is output to both SYSLST and SYSOUT.

LIST = ...,ALL
All the file names on the volume read in are listed. If a save version (svid) has been specified in the FROM operand at the same time, all the files in this save version are listed (not only those specified in the FILES or JOBVAR statement).

PARAM
Set runtime values

The PARAM statement is used to set parameters for the entire ARCHIVE run. The values of these operands are valid up to the END statement unless they are modified by another PARAM statement using the same operands. The values of operands for which no specification is made remain unchanged.
At least one operand must be specified.

The PARAM statement must be entered before the FILES statement. It cannot be continued; if necessary, a further PARAM statement must be issued.

The syntax of the PARAM statement does not conform to the description on page 105. The underscored values indicate the ARCHIVE default values after invocation. If a parameter was already modified in an earlier PARAM statement, this value does not revert to the default value in a subsequent PARAM statement in which this parameter is not specified.

PARAM
CNS = YES / NO ,RESTART = YES / NO ,UNLOAD = NO / YES ,OP[ERATOR] = NO / YES ,WRCHK = NO / YES ,SNR = YES / NO ,DESTROY = NO / YES ,CATID = NO / YES ,OLS = NO / YES

CNS =
This determines whether all files cataloged in the directory file are output in the report.

CNS = YES
All files from the FILES statement that are cataloged in the directory file are listed in the report. This also applies to any files not saved by ARCHIVE because they had not been modified.

CNS = NO
Files which are cataloged but not saved (because they had not been updated) are not to be included in the report.

If the INQUIRE statement is being executed and PARAM `CNS=NO` has been set, only the save versions in which each file was actually saved are output. Output with the qualifier “CNS” is suppressed.

RESTART =
This determines whether checkpoints are written in the checkpoint file.

RESTART = YES
Checkpoints are to be written to the CHKPT file. In the event of the ARCHIVE run being aborted, e.g. due to a system crash, the user can continue the run later on with the aid of a PROCESS statement.

Restrictions on the use of RESTART without a directory file are described on page 82.

RESTART = NO
Checkpoints are not written to the CHKPT file and the user cannot resume the interrupted run.

UNLOAD =
This determines whether magnetic tapes or MT cartridges are unloaded after they have been processed.

UNLOAD = NO
The tapes or MT cartridges are not unloaded at the end of the ARCHIVE run.

UNLOAD = YES
The tapes or MT cartridges are unloaded after they have been processed. During a *restoration run*, every FILES or JOBVAR statement is processed individually. After each processing step the volume is rewound and unloaded. Then the volume is read from the beginning.
During a *save run* the volume is not unloaded until all FILES or JOBVAR statements have been executed.

OPERATOR =
This controls the ARCHIVE messages that require a response from the operator.

OPERATOR = NO
Messages requiring a response from the operator are not output on the console. ARCHIVE initiates a standard error handling routine.

OPERATOR = YES
Messages requiring a response from the operator are output at the operator console. The operator can initiate standard or special processing in the response (see page 94).

WRCHK =
This determines whether a read-after-write check is conducted when writing to hard disk. The read-after-write check is not performed by ARCHIVE but by DMS (see the `/ADD-FILE-LINK` command in the “DMS” manual [2]).

WRCHK = <u>NO</u>
No read-after-write check is performed by DMS.

WRCHK = YES
After each write operation the results are read, thus enabling any errors to be detected immediately, albeit at the cost of increased runtime.

SNR =
This controls the scope of the ARCHIVE report if `REPLACE=NO`.
SNR = Saved Not Restored; files that have been saved but not restored.

SNR = <u>YES</u>
Files which were not written back because they already existed are listed in the ARCHIVE report.

SNR = NO
In restoration runs with `REPLACE=NO`, files which were not written back (because they already existed) are not listed in the ARCHIVE report.

DESTROY =
This specifies what happens to the storage space when a save file is deleted.

DESTROY = <u>NO</u>
The storage space for the save file is returned unchanged to the system on deletion.

DESTROY = YES
If a disk save file created in this run with SAVE or EXPORT is deleted with `/DELETE-FILE` or with the ARCHIVE statement PURGE, the storage space is overwritten with binary zeros (cf. “Commands” [4] manual `/CREATE-FILE`).

CATID =
This determines whether or not the catalog ID should be used (see section “Supporting pubsets” on page 72).

CATID = <u>NO</u>
The files or job variables are accessed according to the standard system conventions.

CATID = YES
This specifies that the catalog ID of the files or the job variables is taken into account. The catalog ID is specified in the file name/job variable name.

OLS =
Controls online saving in UDS databases.

OLS = NO
UDS databases are not saved online.

OLS = YES
UDS databases are saved online, i.e. in an open state. An after-image file must have been created for the database.

POOL
Manage volume pool

Using this statement the user can add/remove tapes or MT cartridges to/from the volume pool of his directory file.

The ADD and REMOVE operands may be specified concurrently, but at least one of them must always be specified.
All messages are output via SYSOUT; no report is produced.

If the volumes are managed via MAREN, they should not be assigned by means of the POOL statement (see section “ARCHIVE and MAREN” on page 55).

PO[OL]
DIR[ECTORY] = ARCHIVE.DIR[,NEW] / filename[,NEW] [,ADD = vsn / (vsn,...) ,DEVICE = TAPE-C4 / device type] [,REM[OVE] = vsn / (vsn,...)] ,NOW = YES / NO[,TIME = sec] [,P[ASSWORD] = password]

DIRECTORY = filename / filename,NEW
This specifies the directory file that is processed with the POOL statement.
If NEW is specified, a file with the specified name is initialized.

ADD = vsn / (vsn,...)
This operand adds VSNs of volumes to the pool of the specified directory file.

DEVICE = TAPE-C4 / device type
This operand is used to define the device type for the VSNs of volumes which are added to the pool.

The “type” entry must conform to the DMS specifications. A volume specified with a certain recording density is used as an output volume even if a different recording density is specified in a SAVE or EXPORT statement.

REMOVE = vsn / (vsn,...)
This operand removes the volume serial numbers of volumes from the pool of the specified directory file. The volumes may not be reserved by any save version at the time of removal.

NOW =
This operand specifies when the POOL statement is to be executed.

NOW = YES
The POOL statement is executed immediately.

NOW = NO
The POOL statement is checked for correct syntax and then written to the queue file ARCHIVE.LATER. This process is given the ARCHIVE identifier L.yymmdd.hhmmss. The identifier is only allocated if no LATER job exists in the same task at the time. If this is not the case, then the process is stored under the existing identifier.
The LATER job can be started by means of a PROCESS statement or deleted by means of a DELETE statement.

NOW = NO,TIME=sec
The number of CPU seconds that are available to this process when it is started. This entry corresponds to the operand CPU-LIMIT in /START-(EXECUTABLE-)PROGRAM.

PASSWORD = password / (password,...)
If you specify the NOW=NO operand, you must enter here the passwords for

- the directory file, if it is protected by a write password and is not under TSOS.
- magnetic tapes and MT cartridges that are protected by a password (up to 4 characters) under MAREN.

In all other cases, you should enter the passwords with the /ADD-PASSWORD command before ARCHIVE is called. Only in this case are the passwords also valid for the subsequent ARCHIVE runs of the same user job.

The system support must specify the password for the directory file if this file is assigned under another user ID.

Example

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*POOL DIRECTORY=ARCHIVE.DIR,NEW,ADD=(OS0230,OS0231),DEVICE=TAPE-C4 ———  (1)
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100813.141905',
           VERSION='9.0A' ————————————————————————————————————————————  (2)
%  MARM121 MAREN CATALOG ENTRY 'OS0230'/'0001' MODIFIED
%  ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'OS0230' ADDED TO THE POOL   (3)
%  MARM121 MAREN CATALOG ENTRY 'OS0230'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'OS0231'/'0001' MODIFIED
%  ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'OS0231' ADDED TO THE POOL   (4)
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) The directory file `ARCHIVE.DIR` is created. The named MT cartridges should be entered in the volume pool of the directory file.

(2) ARCHIVE confirms that the entered statement was syntactically correct. The ARCHIVE run is given an ARCHIVE sequence number.

(3) ARCHIVE reports that MT cartridge `OS0230` has been added to the volume pool of the directory file.

(4) ARCHIVE reports that MT cartridge `OS0231` has been added to the volume pool of the directory file.

PROCESS
Start waiting ARCHIVE runs

With the PROCESS statement, you can start ARCHIVE processes defined with `NOW=NO` or continue ARCHIVE processes already started and then interrupted (e.g. due to a system crash).

PRO[CESS]
archive-seqno

archive-seqno
15-digit ARCHIVE sequence number in the form `L.yymmdd.hhmmss` for processes which are waiting or `A.yymmdd.hhmmss` for interrupted ARCHIVE processes.

The system support may omit this operand. In this case, all ARCHIVE processes defined with `NOW=NO`, which are thus in the queue file `ARCHIVE.LATER`, are started.

PROCESS notes

- Only EXPORT, IMPORT, RESTORE and SAVE runs can be continued by means of the PROCESS statement.
  If, for example, a number of VSNs are removed from the directory file by means of a POOL statement and execution then aborted, you cannot continue this statement; the entire POOL statement must be repeated. This causes a warning message to be issued, as some of the VSNs are now missing from the directory file.
- Restrictions affecting RESTART without a directory file are described on page 82.
- If alias names are specified in commands, the file names are taken from the alias catalog which is active when the batch job is executed.

Example

See page 32.

PURGE
Delete from directory file

This statement permits save versions whose retention period has expired to be deleted from the directory file.

```
PU[RGE]

DIR[ECTORY] = ARCHIVE.DIR / filename

,SV = OBS[OLETE] / svid / (svid,...) / backup,date[.time]
,FORCE = NO / YES

,NOW = YES / NO[,TIME = sec]
[,P[ASSWORD] = password / (password,...)]
,L[IST] = { SYSLST / SYSOUT / BOTH / NONE } [,ALL]
```

DIRECTORY = ARCHIVE.DIR / filename
This specifies the directory file to be processed.

SV =
This defines the save versions to be deleted.

SV = OBSOLETE
All the save versions whose retention period has expired are deleted from the specified directory file.

SV = svid / (svid,...)
This specifies the save version(s) to be deleted. Any save version specified is deleted only if its retention period has expired or if `FORCE=YES` is specified.

SV = backup,date.time
The save versions created with the specified BACKUP entry before or at the time and date given and whose retention period has expired, are deleted from the directory file.

FORCE =
This determines whether save versions that are still valid should be deleted.

FORCE = NO
Only the obsolete save versions are deleted.

FORCE = YES
All explicitly specified save versions are deleted.

NOW =
This operand specifies when the statement is to be executed.

NOW = YES
The PURGE statement is executed immediately.

NOW = NO
The PURGE statement is checked for correct syntax and then written to the queue file ARCHIVE.LATER. This process is given the ARCHIVE identifier L.yymmdd.hhmmss. The identifier is only allocated if no LATER job exists in the same task at the time. If this is not the case, the process is stored under the existing identifier.
The LATER job can be started by means of a PROCESS statement or deleted by means of a DELETE statement.

NOW = NO,TIME=sec
The number of CPU seconds that are available to this process when it is started. This entry corresponds to the operand CPU-LIMIT in /START-(EXECUTABLE-)PROGRAM.

PASSWORD = password / (password,...)
If you specify the operand NOW=NO, you must enter here the passwords for

- the directory file, if it is protected by a write password and is not under TSOS.
- the save version, if it has been protected by a password with SVPASS.
- magnetic tapes and MT cartridges that are protected by a password (up to 4 characters) under MAREN.

In all other cases, you should enter the passwords with the /ADD-PASSWORD command before ARCHIVE is called. Only in this case are the passwords also valid for the subsequent ARCHIVE runs of the same user job.

The system support must enter the password for the directory file if this file is assigned under another user ID.

LIST =
This specifies the output medium. The deleted save version and its associated volumes are output.

LIST = SYSLST
A list of the deleted save versions and tapes is output to SYSLST.

LIST = SYSOUT
A list of the deleted save versions and tapes is output to SYSOUT.

LIST = BOTH
Output is directed to both SYSLST and SYSOUT.

LIST = NONE
The deleted save versions are not output.

LIST = ...,ALL
In addition to the save versions and volumes all the information about the relevant files and job variables is output.

Notes on PURGE

- If the files belonging to a deleted save version are on disk, these files are also deleted. Any tapes which become free and were not entered in the directory file by means of the POOL statement, but were assigned to a save version via the TAPES/VOLUME operand, via MAREN or by interrogation at the console, are purged from the pool.
- Tapes given the status `UNUSABLE` because they could not be opened owing to an unrecoverable write error are not purged from the directory file. The user must remove them from the directory file himself by means of the POOL statement (REMOVE operand).
- An ARCHIVE run with `PURGE ...,FORCE=YES` also deletes save versions whose retention period has not yet expired. Tapes on which this save version is stored are not released in the process. These tapes cannot be used until the expiration date has been reached.
- The following steps must be carried out in order to return a tape whose retention period has not yet expired to the pool:

  1. Delete all the save versions with the statement `PURGE ..., FORCE=YES`.
  2. Delete the tape with the volume serial number vsn from the pool of the specified directory file with the statement `POOL REMOVE=vsn`.
  3. Initialize the tape again with the aid of the INIT utility routine (see the “Utility Routines” manual [1]).
  4. Add the tape with the volume serial number `vsn` to the pool of the specified directory file with the statement `POOL ADD=vsn`.

Example

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*PURGE DIRECTORY=ARCHIVE.DIR.B,SV=OBSOLETE,LIST=SYSOUT ————————————————  (1)
*       P U R G E   C O M M A N D   L I S T I N G ***

PURGE DIRECTORY=ARCHIVE.DIR.B,SV=OBSOLETE,LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100813.145228',
           VERSION='9.0A'
%  MARM121 MAREN CATALOG ENTRY 'OS0427'/'0001' MODIFIED
*           ***        P U R G E   A R C H I V E   R E P O R T  *** ———  (2)
                                    DIRECTORY - $TSOS.ARCHIVE.DIR.B
     SAVE VERSION         VSNS

   S.100813.145211       OS0427
            ***      E N D   O F   A R C H I V E   R E P O R T  ***
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) ARCHIVE is to delete from the directory file ARCHIVE.DIR all save versions whose retention periods have expired and to output the log via SYSOUT.

(2) The ARCHIVE report is output via SYSOUT.
It contains the following information:

- SAVE VERSION: Name of the deleted save version.
- VSNS:
  VSN of the volume which was occupied by the deleted save version. The volume is available again.

RESTORE
Restore from volumes

The RESTORE statement restores files, file blocks, job variables or catalog entries that were saved to tape, MT cartridge or disk in a SAVE run.

The files and/or job variables to be processed have to be specified in FILES and/or JOBVAR statements before the RESTORE statement is entered.

```
R[ESTORE]

DIR[ECTORY] = $TSOS.ARCHIVE.DIR / filename / NONE

,FR[OM] = LATEST[,STATE] / date[.time][,STATE] / svid[,STATE,date[.time]] / [svid,][(vsn,...),...]
,DEVICE = TAPE-C4 / device type
,DRIVES = 1 / integer

,CONSISTENCY-CHECK / CONS-CHK = NO / YES

,REP[LACE] = NO / YES / ALL / ALLP
,SPACE = REORG / KEEP
,CONV[ERSION] = STD / NO / CONV
,ENVIRONMENT-ATTRIBUTES / ENV-ATT = FROM-TARGET / FR-T / FROM-ORIGIN / FR-O

,NOW = YES / NO[,TIME = sec]
[,P[ASSWORD] = password / (password,...)]
,L[IST] = { SYSLST / SYSOUT / BOTH / NONE } [,ALL]
```

DIRECTORY =
The user specifies here whether a directory file is to be used for writing back, and if so which one.

DIRECTORY = $TSOS.ARCHIVE.DIR / filename
The name of the directory file that was used in the save run.

DIRECTORY = NONE
No directory file is to be used for this run. The VSNs are to be taken from the FROM operand.

If a directory file is being used, a file with the same name as the directory file will not be restored in the RESTORE run.

A RESTORE run can also be performed with a systems support directory file under the TSOS user ID if the directory file is password-protected and the password has not been entered.

FROM =
This specifies the input volumes or save version from which the files or job variables are to be read.
If no directory file is used, i.e. `DIRECTORY=NONE` is specified, the VSNs of the volumes must be specified.
If VSNs are specified, the volumes must all be from the same save version and grouped according to the parallel runs. Consequently, the number of parallel runs must not exceed the number of groups of volumes.

FROM = LATEST
Each file or job variable requested is read in from the last save version containing it.
A directory file must be specified for this.

FROM = LATEST,STATE
As for LATEST, but only the files and/or job variables which were saved in the last save run performed with this directory file or which are marked CNS (cataloged but not saved) are read in. Files or job variables which at the time of this save run have already been deleted, or which have been excluded by means of the EXCEPT operand, are not read in.
Specification of a directory file is mandatory.

FROM = date.time
The requested files or job variables are read in from the last save version created before or at this time which contains these files.
Default value for time: `235959`
A directory file must be specified.

FROM = date.time,STATE
As above. Here, however, only those files or job variables are read in which were saved in the very last version of the directory file before or at this time, or which are marked as CNS (Cataloged Not Saved). Files or job variables which have already been deleted or have been excluded by means of the EXCEPT operand are not read in.
A directory file must be specified.

FROM = svid,STATE,date.time
Every file requested in the FILES statement or job variable requested via the JOBVAR statement which is contained in this save version is read in. Files or job variables, however, which were saved again between the specified save version and the time given (both inclusive) are not restored.
date.time is rendered in the format `yymmdd.hhmmss`. Default value for time: `235959`.
A directory file must be specified.

FROM = (vsn,...)
Every required file or job variable on the specified volumes is written back.
The volumes must be listed in the order in which they were recorded during the save run.
If a directory file is used, only those files located in their entirety on the specified volumes are read in.

FROM = (vsn,...),(vsn,...)
When files or job variables saved in parallel runs are restored, the volumes must be grouped according to the parallel runs, i.e. the volumes of a parallel run must be specified in each parentheses in the order in which they were recorded.

Data which was saved using

```
FILES
SAVE ...,DRIVES=2,TAPES=(vsn1,vsn2),
```

and distributed according to the EXPORT listing in two parallel runs over vsn1 and vsn2, must be restored using

```
FILES
RESTORE ...,DRIVES=2,FROM=(vsn1),(vsn2).
```

A restoration run implemented via

```
RESTORE ...,DRIVES=2,FROM=(vsn1,vsn2)
```

causes the run to abort, as ARCHIVE cannot allocate the volumes correctly.

FROM = svid
All the required files or job variables contained in the specified save version are restored. A directory file must be specified.

FROM = svid,(vsn,...),...
As above.
In addition, if a directory file has been specified, a check is made to see whether the specified volumes belong to this save version. If this is not the case, the run is aborted.

DEVICE = TAPE-C4 / device type
The device type is specified via this operand. It must be the same as the one used for the save run. The DEVICE operand defines the device type for all the VSNs, including those specified indirectly, e.g. in `FROM=svid`.
If a directory file is specified, the device type is taken from the save version of the directory file.

DRIVES = 1 / integer
This specifies the number of parallel runs (up to 16). As many parallel runs should be specified as were defined for the associated SAVE run.

CONSISTENCY-CHECK / CONS-CHK =
This determines whether a consistency check of the save data is carried out.

CONSISTENCY-CHECK = NO
No consistency check is carried out.

CONSISTENCY-CHECK = YES
Before transferring save data, the check bytes are calculated and a comparison is made with the check bytes stored with the save data in EXPORT. If an error occurs, the message ARC0413 is output.

This operand value is ignored if the EXPORT or SAVE operations were carried out with the CONSISTENCY-CHECK=NO operand.

REPLACE =
This specifies whether or not existing files and job variables with the same names are to be replaced.

REPLACE = NO
A file or job variable is not read in if one already exists with the same name; in this case a message is issued.

REPLACE = YES
If a file or job variable with this name exists, it is replaced by the file or job variable from the save run.

However, a file or job variable is not replaced if:

– only read access is permitted for it
– it has an expiration date (EXPIR DATE) in the catalog which is later than the current date
– it is protected by a password which has not been specified.

REPLACE = ALL
If a file or job variable with the same name already exists, it is replaced by the saved file or job variable. The file or job variable is replaced even if only read access is permitted for it or the expiration date has not yet been reached. However, if the existing file is protected by a password, this password must be specified.

REPLACE = ALLP
This operand can only be specified under the ID of the system support. It enables files or job variables protected by passwords to be replaced by the corresponding files from the save run, without the passwords having previously been specified. Otherwise the same applies to the ALLP operand as to ALL.

SPACE =
This specifies whether or not the files which are to be replaced by files with the same names from the save run are to be deleted before restoration.
The operand is not interpreted for migrated files.

SPACE = REORG
Files to be replaced by files with the same names from the save run are deleted before restoration. Usually the file then occupies a different extent after restoration. Thus restoration runs with SPACE=REORG have the effect of reorganizing the disks.

SPACE = KEEP
Files to be replaced by files with the same names from the save run are overwritten in the extents which they occupy at the time of the restoration.
The specification SPACE=KEEP overrides any definitions of the storage area for the file being restored.

Example

The file HUGO is to be written back with SPACE=KEEP. Let us assume that there is already a file of this name on the private disk PRIV01. If TO=PUBLIC is then specified for this run in the FILES statement, ARCHIVE still writes the file back to the private disk PRIV01. In other words the specification SPACE=KEEP overrides the specification TO=PUBLIC.

The specifications in the SPACE operand are ignored if REPLACE=NO has also been specified.

The specifications in the SPACE operand are ignored for file generation groups. FGGs are deleted before writing back takes place.

The following points must be borne in mind for ISAM files with a separate file index section and file data section:

– The file in the system must have the same logical structure as the saved file, i.e. the index blocks and data blocks must occupy the same logical blocks.

– The access method (ACCESS-METHOD) must be ISAM.

The SPACE operand is irrelevant for catalog entries for tape files. When REPLACE=YES is specified, catalog entries are always deleted.

CONVERSION =
This operand controls whether files which had a PAM key when they were saved must be converted if the output volume is an NK4 disk.
This operand is ignored for job variables.
Entering CONVERSION=STD / CONV requires the PAMINT subsystem.
In the case of partially saved files, only CONVERSATION=NO / CONV may be specified.
If a conversion file which is to be imported already exists, it is always reorganized; the SPACE operand is ignored.

CONVERSION = STD
In RESTORE runs to an NK disk, files with PAM keys are converted by means of PAMINT in accordance with the following rules:

– K-ISAM files to NK-ISAM files
(BLOCK-CONTROL-INFO=WITHIN-DATA-BLOCK)

– K-SAM files to NK-SAM files
(BLOCK-CONTROL-INFO=WITHIN-DATA-BLOCK)

– K-UPAM files to NK-UPAM files
(BLOCK-CONTROL-INFO=NO)

PAM files whose structure is known (load modules, libraries) are converted accordingly (e.g. LMR libraries to PLAM libraries). In the case of PAM files whose structure is not known, the PAM key information is lost. If the PAM key contained any information, an appropriate message is displayed.

CONVERSION = NO
Files with a PAM key are not converted in RESTORE runs to an NK disk and are thus not restored.

CONVERSION = CONV
In a RESTORE run to an NK disk, files with PAM keys are converted to “CONV format” by means of PAMINT. The restored file contains all the PAM keys at the end of the file in separate blocks.

If a file has to be converted in a RESTORE run, the SPACE operand has no effect: the file to be replaced is deleted before it is restored.

ENVIRONMENT-ATTRIBUTES =
This specifies how the attributes of files and job variables protected by BACL or GUARDS are to be processed during a RESTORE for a save file (see section “File attribute handling” on page 88). This save file was created with SAVE with simultaneous renaming of the catalog ID or user ID.

ENVIRONMENT-ATTRIBUTES = FROM-TARGET
The attributes for access protection are reset to match the status of the target file or job variable.

ENVIRONMENT-ATTRIBUTES = FROM-ORIGIN
The attributes for access protection are taken from the save file. If the user ID has been renamed, this value can only be specified by a user with the TSOS privilege.

NOW =
This operand specifies when the statement is to be executed.

NOW = YES
The RESTORE statement is executed immediately.

NOW = NO
The RESTORE statement is checked for correct syntax and then written to the queue file ARCHIVE.LATER. This process is given the ARCHIVE identifier L.yymmdd.hhmmss. The identifier is only allocated if no LATER job exists in the same task at the time If this is not the case, then the process is stored under the existing identifier.
The LATER job can be started by means of a PROCESS statement or deleted by means of a DELETE statement.

NOW = NO,TIME=sec
The number of CPU seconds that are available to this process when it is started. This entry corresponds to the operand CPU-LIMIT in /START-(EXECUTABLE-)PROGRAM.

PASSWORD = password / (password,...)
If you specify the NOW=NO operand, you must enter here the passwords for

- files or job variables to be imported which are protected by read passwords.
- the directory file, if it is protected by a write password and is not under TSOS.
- the save version, if it has been protected by a password with SVPASS.
- magnetic tapes and MT cartridges that are protected by a password (up to 4 characters) under MAREN.

In all other cases, you should specify the passwords with the /ADD-PASSWORD command before you call ARCHIVE. Only in this case are the passwords also valid for the subsequent ARCHIVE runs of the same user job.

The system support can process files and job variables without specifying the passwords. The password for the directory file, however, must be specified if the directory file is assigned under another user ID.

LIST =
This defines the output medium for the report of this ARCHIVE run.

LIST = SYSLST
The report is output to SYSLST.

LIST = SYSOUT
In interactive mode, the report is output to the terminal.

LIST = BOTH
The report is output to both SYSLST and SYSOUT.

LIST = NONE
No report is generated.

LIST = ...,ALL
All the file names on the volumes read in are listed. If a save version (svid) is also specified in the FROM operand, all the file names belonging to this save version are listed (not only those specified in the FILES/JOBVAR statements).

Notes on RESTORE

If a requested file that has only been partially saved (operand `CHANGED=YES,LARGE` or `CHANGED=YES,integer` in the save run), ARCHIVE proceeds in the following manner:

- If the operand `FROM=LATEST[,STATE]` or `FROM=date.time[,STATE]` is specified, ARCHIVE will automatically first read the whole file from the full save associated with the partial save and then the modified pages from the partial save.
- Any other entry in the FROM parameter causes ARCHIVE to read only the partial save for such a file, i.e. the pages modified since the associated full save. Consequently, in order to obtain correct results, the user himself must ensure that this file exists prior to the RESTORE run and that it has the status that was current at the time of the associated full save.
- The restore of a file is rejected with an error message, if the corresponding user ID is missing.

  With an optional rep the missing user ID is created for an administrator restore and the file gets restored.

SAVE
Save to volumes

With the aid of the SAVE statement, you can save the following data on tape, MT cartridge or disk:

- catalog entries of files on private volumes
- entire files on private or public disks
- file blocks which have been modified since the last full save
- job variables

Save data created in an earlier SAVE run can be copied to these volumes. The saves produced by SAVE can be read in again in a RESTORE run.

The files and/or job variables to be processed must be specified in FILES and/or JOBVAR statements before you enter the SAVE statement.

S[AVE]

DIR[ECTORY] = ARCHIVE.DIR[,NEW] / filename[,NEW] / NONE

,DIRSAVE = NO / YES
,BACKUP = E / D / C / B / A
,CH[ANGED] = YES[,LARGE / PAM pages] / NO
,UNUSED = 0 / days
,CAT[ONLY] = NO / YES
,DUP[LICATE] = YES / NO

,S[AVE]-ACL = YES / NO

[,CONT[INUE] = svid / [svid,](vsn,...),...]
,TAPES / VOLUME = PO[OL] / OP[ERATOR] / vsn / (vsn,...)
[,MAREN-LOCATION = <alphanum-name 1..8>]
,DEVICE = TAPE-C4 / device type
,DRIVES = 1 / integer

,RETPD = 0 / days
,SH[ARE] = NO / YES
[,SVPASS = password]
,PRIM[ARY] = 500 / PAM pages
,SEC[ONDARY] = 100 / PAM pages
[,BL[OCK-SIZE] = STD / MAX / PAM-Seiten]
,COMP[RESS] = NO / YES
,CONSISTENCY-CHECK / CONS-CHK = NO / YES

,ER[ASE] = NO / YES / ALL / ALLP / ALLCNS / ALLPCNS

,NOW = YES / NO[,TIME = sec]
[,P[ASSWORD] = password / (password,...)]
,L[IST] = SYSLST / SYSOUT / BOTH / NONE

DIRECTORY =
This specifies whether a directory file is to be used for the SAVE run, and if so which one. The person initiating the run must be authorized to read and write in this directory file. Any necessary passwords must be entered with the `/ADD-PASSWORD` command.

DIRECTORY = ARCHIVE.DIR / filename
This specifies the name of the directory file to be used in this SAVE run.
The file must already exist and contain entries from ARCHIVE.

DIRECTORY = filename,NEW
A file with this name is to be initialized as a directory file.
NEW must be specified even if the file already exists but is still empty, i.e. contains no entries from ARCHIVE.

DIRECTORY = NONE
No directory file is to be used for this run.

DIRSAVE =
This operand determines whether the directory file is saved to the volume.

DIRSAVE = NO
The directory file is not saved to the volume.
All files which have the same name as the directory file being used are skipped.
When copying a save file to another save file, any file in the input save version with the same name as the directory file will not be copied.

If the directory file is contained in the list of file names to be processed and `DIRSAVE=NO` is specified, the directory file will not be processed.

DIRSAVE = YES
The directory file is saved to the volume as the last file of this run.
The DIRSAVE operand only applies if files are found which are to be saved.

The entire directory file is saved, regardless of the values of the CHANGED and BACKUP operands in the SAVE statement or the corresponding catalog entries.

A directory file is always resident on a single volume, i.e. there are no continuation volumes within a directory file.

A saved directory file cannot be restored if you specify a directory file with the same name in the RESTORE run.

If the directory file is contained in the list of file names to be processed and DIRSAVE=NO is specified, the directory file will not be processed.

BACKUP = E / D / C / B / A
This allows you to specify the files to be saved on the basis of the BACKUP entry in the file catalog.
Only those files are saved whose BACKUP entry in the file catalog is equal to or less than the BACKUP value in the SAVE statement (A < B < C < D < E).

Files which are not to be backed up should have the BACKUP entry E in the file catalog. But files with the BACKUP entry E can also be saved using SAVE. However, this procedure is not recommended.
This operand is ignored for job variables.

CHANGED =
specifies which files are to be fully, partially or not saved.

CHANGED = YES
Only those files are fully saved which have either not been cataloged in the directory file or which have been cataloged there but have been updated since the last full or incremental save.
Incremental saves from a private disk with the status “write lockout” (PPD) cannot be executed.

CHANGED = YES,LARGE
Of the files marked in the catalog as LARGE only the pages modified since the last full save are saved (partial save). If there has been no change, the file is not saved. If there is no full save available, or no full save appropriate to the last partial save, the file is saved completely.

CHANGED = YES,PAM pages
This has a similar effect to that of `CHANGED=YES,LARGE` but is only valid for files which occupy more pages than are given by `PAM pages`. `PAM pages` must be an integer ≥0.

CHANGED = NO
The specified files are always saved completely.

With `CHANGED=YES` a directory file must be specified. If `DIRECTORY=NONE` is specified, `CHANGED=NO` must also be specified.
If too many partial saves are performed between two full saves, ARCHIVE automatically performs a full save.
This operand is ignored when saving job variables.

UNUSED = 0 / days
This allows you to select the files to be saved on the basis of the ACCESS-DATE entry in the file catalog. Files are only saved if they have not been accessed for the period specified. Files that have no ACCESS-DATE entry are not saved.
In the case of a RESTART run, the period is reckoned from the date of the RESTART run. `days` can be a value from 0 - 36159 (99 years). This operand is ignored for job variables.

CATONLY =
specifies whether only catalog entries of the files are saved.

CATONLY = NO
The files and their catalog entries are saved.

CATONLY = YES
Of the files specified in the FILES statement, only the catalog entries for files located on private volumes are saved.
CATONLY=YES presupposes DIRECTORY=NONE, i.e. the latter is assumed even if a directory file is specified.
When job variables are saved, this operand is ignored.

Catalog entries saved with CATONLY=YES are restored with the value of the file size at the time of the save run.

DUPLICATE =
This specifies whether files or job variables that have been entered in several FILES or JOBVAR statements are also saved several times.

DUPLICATE = YES
Files or job variables which exist more than once are to be saved more than once; a warning is issued when this happens.

DUPLICATE = NO
All duplicate files or job variables are to be saved only once, i.e. the first file or job variable specified in the FILES or JOBVAR statement is processed.

DUPLICATE=NO is only permitted if a directory file is specified.

SAVE-ACL =
This determines whether the ACL entries (SECOS up to V3.0) for saved files are also saved.

SAVE-ACL = YES
The ACL entries of the saved files are also saved.

The value of the BLOCK-SIZE operand must be at least 3.

The save files created with SAVE-ACL=YES cannot be read by the software product SIR.

SAVE-ACL = NO
The ACL entries of the saved files are not saved.

CONTINUE =
This operand determines that an existing save version is to be extended.
The last volume of this save version is requested again. The new files or job variables are written after the last file or job variable.

If the user has specified a directory file for the save run to be continued, ARCHIVE does not save files which

- have already been saved in a later save version of the specified directory file, or
- have already been saved in the save version which is being continued.

ARCHIVE includes files not saved for these reasons in the report. This ensures that

- no file is saved more than once under the same svid
- a subsequent RESTORE run with FROM=LATEST produces the correct results.

Any extra continuation volumes required for the output must be specified in the TAPES/VOLUME operand.

The first volume, i.e. the one to be continued, may not be specified in the TAPES/VOLUME operand. It should be specified in the CONTINUE operand or it will be found automatically.

The DIRECTORY, CATONLY, SAVE-ACL, BACKUP, CHANGED, COMPRESS, CATID, SVPASS and BLOCK-SIZE operands must be assigned the same values as the corresponding operands of the run to be continued.
If the operands DIRSAVE, DEVICE, ERASE, LIST or SHARE do not have the same values as they had for the first run, the warning `ARC0055` is issued.

The RETPD operand is ignored in a CONTINUE run. The new retention period is calculated internally in such a way that the expiration date matches that of the original save.

If tapes or MT cartridges created with an ARCHIVE version < V2.8A do need to be continued for some reason, they have to be previously converted to the new format via tape-to-tape saving (see the FROM operand in the FILES statement).

In runs with `DIRECTORY=NONE` the user must ensure that files with the same names are not saved in the same save version, since it will otherwise be impossible to predict which of the files will actually be restored in a subsequent RESTORE run.

For save versions on disk, the specification `svid,(vsn,...)` is mandatory, even in runs without a directory.
The continuation disks are specified in the VOLUME operand. They must be of the same device type.
If no continuation disk is required, the VOLUME specification can be omitted. When the specified disks are full, ARCHIVE aborts the run. Sufficient disks must therefore be specified in advance.

If DIRSAVE=YES is entered together with the CONTINUE operand and the original save run was also performed with DIRSAVE=YES, the message ARC0040 is output for the directory file.

CONTINUE = svid
A directory file must have been specified. The last VSN is taken from the directory file for each parallel run.

CONTINUE = svid,(vsn,...)
If, for this combination of operands, a directory file is also specified, ARCHIVE checks whether the specified volumes belong to this save version (svid):

– if this is not the case, ARCHIVE does not execute the save run;
– if it is the case, the last volume is requested and continued.

If no directory file is specified, ARCHIVE requests only the last volume and checks whether it belongs to the specified svid. If it does, the save version is continued.

Regardless of whether or not a directory file is used, it is sufficient to specify the VSN of the last volume.

CONTINUE = (vsn,...),...
This combination of operands is permitted only for tapes or MT cartridges.
If only the VSNs are specified and a directory file is used, ARCHIVE checks whether the volumes all belong to the same save version (svid). If any of the volumes does not belong to this svid, ARCHIVE does not perform the save run.

The specification of VSNs or VSN groups has no effect on the allocation of volumes to the subtask. The continuation volumes are allocated to the subtask automatically in this case. The allocation corresponds to that of the original save run.

If no directory file has been specified, no check is carried out.

Regardless of whether or not a directory file has been specified, it is sufficient to specify the VSN of the last volume.

TAPES / VOLUME =
This operand enables the VSNs of the output volumes to be defined.
POOL and OPERATOR are permitted only for tapes and MT cartridges.
POOL only has meaning when together with a directory file; otherwise `TAPES=OPERATOR` assumed.

TAPES / VOLUME = POOL
The VSNs are taken from the volume pool of the specified directory file in ascending order. If additional volumes are required or if no directory file has been specified, the volumes are requested from the Operating or by MAREN.
After deletion of the save version, the volumes are again AVAILABLE in the pool.
This operand is not permitted when saving to disk.

TAPES / VOLUME = OPERATOR
If MAREN is being used, the volume is made available by MAREN. Otherwise the name of a VSN is requested via message `DMS0591` at the console. The specified VSN is entered in the pool with the qualifier “OPERATOR”.
When the save version is deleted, the volumes are removed from the pool.
This operand is not permitted when saving to disk.

TAPES / VOLUME = vsn / (vsn,...)
The volumes are requested in the specified order.

For tapes and MT cartridges:
The specified VSNs are entered in the pool with the qualifier OPERATOR. When the save version is deleted, the volumes are removed from the pool.
Additionally required volumes are assigned by MAREN or by the operating.

For disks:
If the disks specified for saving are insufficient, ARCHIVE aborts the run.
Only the system support is authorized to specify a public disk, and this must be a disk from the home pubset.

`vsn` may not begin with `PO` or `OP` (conflict with the operand values `OPERATOR` or `POOL`).

DEVICE = TAPE-C4 / device type
This operand specifies the device type for all the VSNs specified in the TAPES/VOLUME or CONTINUE operand. This also applies to VSNs only specified indirectly, e.g. in `CONTINUE=svid`.

MAREN-LOCATION=<alphanum-name 1..8>
This operand specifies the MAREN storage location from which free volumes should be taken if no volume is specified in the statement and no volume is available in the ARCHIVE directory pool.

If the MAREN software product is not installed or started, the operand has no effect. If the operand is specified, it takes precedence over the selection criteria as defined in the MAREN exits.

DRIVES = 1 / integer
This operand specifies the number of parallel runs (up to 16). A number of devices are used in parallel. The value of `integer` must be equal to or less than the number of devices available.

RETPD = 0 / days
This operand defines a retention period in days for the save version. During this period

– the save version in the directory file is protected against being deleted (PURGE),
– tapes and MT cartridges are protected against being overwritten and
– the save files on disk are protected against being overwritten or deleted.

The maximum permissible retention period is 32767 days, i.e. approx. 90 years.

SHARE =
This operand defines whether the volumes or save files that have been created are given the attribute “shareable”.

SHARE = NO
The volumes or save files created are nonshareable. The saved files and job variables can only be read in again under the user ID under which they were saved.
The user cannot even read in his/her files and job variables from a save file of the system support. Other users cannot access the volumes.

SHARE = YES
The volumes or save files created are shareable. The saved files and job variables under the user's ID can be read in. On the other hand, it is not possible to access files and job variables of other users, not even via LIST. Other users can, however, access the volumes themselves.

The specification is only of significance for the system support. If the system administrator carries out system backup using `SHARE=YES`, each user can restore his/her files and job variables from the save file.

SVPASS = password
This operand defines a read password for the save file.
When a file or job variable is to be read in again, this password must be specified either in the PASSWORD operand or via the /ADD-PASSWORD command.

PRIMARY = 500 / PAM pages
This operand specifies the number of blocks (2048 byte) for the primary allocation of the save file (file to which the data is saved), when saving to hard disk.
PAM pages must be an integer between 32 and 50331645.
When saving to tape or MT cartridge, this operand is ignored.

SECONDARY = 100 / PAM pages
This specifies the number of blocks (2048 Byte) for the secondary allocation of save files when saving to hard disk. PAM pages must be 0 or an integer between 32 and 32767. If the value 0 is assigned and ARCHIVE tries to extend the save file, a DMS error occurs.
When saving to tape or MT cartridge, this operand is ignored.

BLOCK-SIZE =
This operand specifies the maximum number of PAM pages of the saved file that are stored in one ARCHIVE save block. BLOCK-SIZE=16, for example, means that up to 16 PAM pages from the saved file are written into one save block.

The operand name TAPE-BLOCK-SIZE is supported only for reasons of compatibility.

If this parameter is not specified, ARCHIVE uses the setting in the ARCHIVE parameter BLOCK-SIZE-T-C for tape cartridge devices and the setting in the ARCHIVE parameter BLOCK-SIZE-TAPE for long tapes.

BLOCK-SIZE = STD
The compatible block size of 32 kB, which corresponds to BLOCK-SIZE=15, is used.

BLOCK-SIZE = MAX
The maximum block size possible in this BS2000/OSD version and for this device is used, in other words 256 kB for BS2000/OSD V6.0 or higher and TCs of type TAPE-C3 or higher, and otherwise 32 kB. When saving to public disk ARCHIVE determines the BLOCK-SIZE on the basis of the disk's maximum transfer length.

BLOCK-SIZE = PAM-pages
The permissible value range depends on the save operation; it extends from 2 to 35:

– The minimum value is 2 for SAVE-ACL=NO or 3 for SAVE-ACL=YES.
– The maximum value is 15 when saving to tape or MT cartridge, 31 when saving to private disk, and 35 when saving to public disk.

For values greater than 31, the user must check their device configuration (with /SHOW-MASTER-CATALOG-ENTRY ...,INFORMATION=*USER) in order to ensure whether such values are permissible.

When saving NK4 disks, only odd values are used; if an even value is specified, a warning is issued and the value is decremented to the next lower odd value.

The default value is 15 for magnetic tape or MT cartridge devices and 31 for hard disk devices. On a hard disk device, values lower than 15 will reduce the performance of ARCHIVE.

COMPRESS =
This operand determines whether the data should be written into the save file in a compressed form.

COMPRESS = <u>NO</u>
The data is not compressed before it is written to the save file.

COMPRESS = YES
The data is written to the save file after software compression.
In the case of device types which automatically execute hardware compression (e.g. TAPE-C4), COMPRESS=NO is assumed.
The save files created via COMPRESS=YES cannot be read by the software product SIR.

CONSISTENCY-CHECK / CONS-CHK =
This operand determines whether check bytes are calculated for a consistency check of the save files.

CONSISTENCY-CHECK = <u>NO</u>
Check bytes are not calculated.

On a subsequent RESTORE or LIST operation, a CONSISTENCY-CHECK=YES entry is ignored.

CONSISTENCY-CHECK = YES
Check bytes are calculated and stored with the save data.
On a subsequent RESTORE or LIST operation, the correct transmission of save data is checked.

ERASE =
This operand determines whether the files and job variables are deleted after a save run.

ERASE = <u>NO</u>
The files and job variables are not deleted after the save run.

ERASE = YES
The saved files and job variables are deleted after the save run if they are not protected by means of a password or a retention period and if modifying access is permissible for them.

ERASE = ALL
As for YES, but those files and job variables which are protected by means of a retention period and for which only read access is permitted are also deleted.

ERASE = ALLP
This operand can only be specified with the TSOS privilege.
As for ALL, but those files and job variables protected by a password are also deleted, without the password having to be specified.

ERASE = ALLCNS
As for ALL, but those files not saved in the current ARCHIVE run (as they were not updated) are also deleted (entry in the directory file: CNS).

ERASE =ALLPCNS
This operand can only be specified with the TSOS privilege.
The effect is as for ALLP, but those files not saved in the current ARCHIVE run (as they were not updated) are also deleted (entry in the directory file: CNS).

File generation groups (FGGs) and file generations are only deleted if the entire FGG was saved in this run.

If the ERASE operand is specified together with `CATONLY=YES`, only the catalog entry is deleted, not the file.

When copying save data, the ERASE operand is ignored.

NOW =
This operand specifies when the save run is to be performed.

NOW = <u>YES</u>
The SAVE statement is executed immediately.

NOW = NO
The SAVE statement is checked for correct syntax and then written to the queue file `ARCHIVE.LATER`. This process is given the ARCHIVE identifier `L.yymmdd.hhmmss`. The identifier is only allocated if no LATER job exists in the same task at the time. If this is not the case, then the process is stored under the existing identifier.
The LATER job can be started by means of a PROCESS statement or deleted by means of a DELETE statement.

NOW = NO,TIME=sec
This specifies the number of CPU seconds available to this process once it is started. This specification corresponds to the CPU-LIMIT operand in the `/START-(EXECUTABLE)-PROGRAM` command.

PASSWORD = password / (password,...)
If you specify the operand NOW=NO, you must enter here the passwords for

- files or job variables to be imported which are protected by read passwords.
- the directory file, if it is protected by a write password and is not under TSOS.
- the save version, if it has been protected by a password with SVPASS.
- magnetic tapes and MT cartridges that are protected by a password (up to 4 characters) under MAREN.

In all other cases, you should specify the passwords with the /ADD-PASSWORD command before calling ARCHIVE.

The systems support staff can process files and job variables without specifying the passwords. The password for the directory file, however, must be specified if the directory file is assigned to another user ID.

LIST =
This specifies the output medium for the report of the SAVE run.

LIST = SYSLST
The report is output to SYSLST.

LIST = SYSOUT
In an interactive task, the report is output to the terminal.

LIST = BOTH
The report is output to both SYSLST and SYSOUT.

LIST = NONE
No report is generated.

Notes to SAVE

In the SAVE run, ARCHIVE checks the MT cartridge entries. If a DMS message appears (e.g. wrong entry), it may not be answered with tsn.IGNORE. Otherwise there is the danger that the volume archive number saved in the directory file will not match the actual one. This will lead to diffictulties in a later RESTORE run.

SHOW-DEFAULT
Display the default parameter settings

You can use the SHOW-DEFAULT statement to display the current default values of the ARCHIVE parameters, i.e. the values stored in the ARCHIVE parameter file when ARCHIVE was started.

SH[OW]-DEF[AULT]
[L[IST] = SYSOUT / SYSLST / BOTH]

LIST =
specifies where the output is to be sent.

LIST = SYSOUT
The report is output on the data display terminal in interactive mode.
This is the default for runs in interactive mode.

LIST = SYSLST
The report is output on SYSLST.
This is the default for runs in batch mode.

LIST = BOTH
The report is output on SYSLST and SYSOUT.

Table of ARCHIVE parameters in the ARCHIVE parameter file

The following table contains all ARCHIVE parameters in alphabetical order and

- the statements in which the ARCHIVE parameters are present
- the permissible values for each ARCHIVE parameter
- the presetting in the ARCHIVE parameter file (see page 23)

Parameter	Statement	Permissible values/syntax	Presetting
BACKUP	SAVE	A / B / C / D	D
BLOCK-SIZE-DISK	SAVE, EXPORT	STD / MAX / <integer 1..35>	MAX
BLOCK-SIZE-T-C	SAVE, EXPORT	STD / MAX / BIG / <integer 1..15>	BIG
BLOCK-SIZE-TAPE	SAVE, EXPORT	STD / MAX / <integer 1..15>	MAX
CATID	PARAM	NO / YES	NO
CHANGED	SAVE	NO / YES / YES,LARGE / YES,<integer 1..2^{31}>	YES
CHECK-PUB-SPACE	—	NO / YES	NO
CNS	PARAM	NO / YES	YES
COMPRESS	SAVE, EXPORT	NO / YES	NO
CONSISTENCY-CHECK	SAVE etc.	NO / YES	NO
DESTROY	PARAM	NO / YES	NO
DEVICE-TAPE-T-C	SAVE etc.	<device type>	TAPE-C4
DIR-E-I-L	EXPORT, IMPORT, LIST	NONE / <filename 1..41> (no catid and no userid)	NONE
DIR-R-Q	RESTORE, INQUIRE	NONE / <filename 1..47> (no catid and no userid or no catid and userid = TSOS)	$TSOS.ARCHIVE.DIR
DIR-S-PO-PU	SAVE, POOL, PURGE	NONE / <filename 1..41> (no catid and no userid)	ARCHIVE.DIR
DIRSAVE	SAVE, EXPORT	NO / YES	NO
DUPLICATE	SAVE, EXPORT	NO / YES	YES
ERLIST	—	NO / YES	NO
FGERASE	—	NO / YES	NO
LIST-INQUIRE	INQUIRE, SHOW-DEFAULT	BOTH / SYSLST / SYSOUT / TASK-TYPE-DEPENDING	TASK-TYPE-DEPENDING
LIST-OTHERS	SAVE etc.	NONE / BOTH / SYSLST / SYSOUT	SYSLST

Parameter	Statement	Permissible values/syntax	Presetting
NOW	SAVE etc.	NO / YES	YES
OLS	PARAM	NO / YES	NO
OPERATOR	PARAM	NO / YES	NO
PRIMARY	SAVE, EXPORT	<integer 1..50331645>	500
RESTART	PARAM	NO / YES	YES
RETPD	SAVE, EXPORT	<integer 0..32767>	0
SAVE-ACL	SAVE	NO / YES	YES
SECONDARY	SAVE, EXPORT	<integer 0..32767>	100
SHARE-EXPORT	EXPORT	NO / YES	YES
SHARE-SAVE	SAVE, EXPORT	NO / YES	NO
SHUPDW	——	<integer 1..9999>	600
SNR	PARAM	NO / YES	YES
SPACE	RESTORE, IMPORT	KEEP / REORG	REORG
STREAM	PARAM	NO / YES	YES
TARGET-ARCHIVE-VERSION	EXPORT	FROM-V26B / BEFORE-V26B	FROM-V26B
TIME	SAVE etc.	<integer 1..32767>	0
TYPE	STATUS	ACTIVE / LATER	LATER
UNLOAD	PARAM	NO / YES	NO
UNUSED	SAVE, EXPORT	<integer 0..36159>	0
VOLUME	SAVE, EXPORT	POOL / OPERATOR	POOL
WRCHK	PARAM	NO / YES	NO

The ARCHIVE parameter file also contains the following internal parameters, which are reserved for special customer solutions and service personnel. This parameters may only be changed after prior arrangement with the service department.

Internal parameter	Possible values/syntax	Presetting	Note for service
DEADLOCK-WAIT-TIME	0 / <integer 180..32767>	0	see A0475858
PVS-CATID	*NONE / <catid>	*NONE	see A0475856
HSMS-SEC-WAIT-TIME	<integer 0..99999>	0	see A0475855
CUST-PRIO-POSIX	<integer 0..2552	0	see A0449431 and A0335173
CUST-PRIO-CC	<integer 0..255>	0	see A0449445
CUST-PRIO-BS2000	<integer 0..255>	0	see A0409760

Meaning of ARCHIVE parameters

BACKUP
This parameter allows you to specify the files to be saved, on the basis of the BACKUP entry in the file catalog, if the operand of the same name was not specified in the SAVE statement. Only files whose BACKUP entry is less than or equal to the BACKUP parameter value will be saved; thus A < B < C < D. Files with the BACKUP value E cannot be saved with SAVE. The parameter is ignored for job variables.

BLOCK-SIZE-DISK
When saving, this parameter specifies the maximum number of PAM pages per save block for a disk save file if the BLOCK-SIZE operand was not specified in the SAVE or EXPORT statements.

If the save file is to be transferred from one pubset to another, the specified block size must have the same value as the smallest “maximum I/O length -1” of the relevant pubsets.

STD (which corresponds to the value 31) is compatible for all disks.

BLOCK-SIZE-T-C
When saving, this parameter specifies the maximum number of PAM pages per save block for a save file on an MT cartridge if the BLOCK-SIZE operand was not specified in the SAVE or EXPORT statements.

By default BLOCK-SIZE-T-C=BIG, i.e. the maximum block size of 256 kB (for BS2000/OSD V6.0 or higher and TAPE-C3), is used.
For devices which do not support this block size (long tape, TAPE-C1/-C2) ARCHIVE automatically uses the compatible block size of 32 kB, which corresponds to BLOCK-SIZE=15.
For devices which were introduced in BS2000/OSD V6.0 or higher, ARCHIVE has always used a block size of 256 kB.

With continuation of a save file, a selected maximum block size is reduced to the actual block size of the save file. If the selected block size is smaller than the actual block size of the save file to be continued, then the request will be rejected.

In the case of alternating save operations or exchange of volumes between BS2000/OSD V6.0 or higher and earlier versions, only block sizes up to 32 kB can be used. For this purpose BLOCK-SIZE-T-C=STD should be set in the ARCHIVE parameter file.

BLOCK-SIZE-TAPE
When saving, this parameter specifies the maximum number of PAM pages per save block for a save file on magnetic tape if the BLOCK-SIZE operand was not specified in the SAVE or EXPORT statements. BLOCK-SIZE-TAPE=STD uses the compatible block size of 32 kB, which corresponds to BLOCK-SIZE=15.

CATID
This parameter determines whether ARCHIVE should also save the catalog ID for saved files and job variables (see also the section “Supporting pubsets” on page 72).

CHANGED
This parameter specifies which files are to be fully, partially or not saved if the operand of the same name was not specified in the SAVE statement.

CHECK-PUBSET-SPACE
This parameter specifies whether ARCHIVE should check the pubspace limit for SAVE/EXPORT operations to disk and for RESTORE statements and if necessary cancel the operation. It checks whether the assigned storage space has been exceeded for a nonprivileged user and whether level 4 of the storage space saturation has been reached.

CNS
This parameter (Cataloged Not Saved) specifies whether all files cataloged in the directory file are output in the report, even if they have not been saved because they contain no changes.

COMPRESS
This parameter determines whether files are to be written to the save file using software compression if the operand of the same name was not specified in the SAVE or EXPORT statements.

CONSISTENCY-CHECK
This parameter determines whether check bytes are calculated and stored for every save block in SAVE/EXPORT statements and whether the check bytes should then be checked in the IMPORT/RESTORE and LIST statements if the operand of the same name has not been specified in these statements.

DESTROY
This parameter specifies whether the storage space occupied by a disk save file should be overwritten with binary zeros when the file is deleted.

DEVICE-TAPE-T-C
This parameter specifies the device type for the VSNs of the statement (MT cartridge or magnetic tape) if the DEVICE operand was not specified in the respective statement (SAVE, etc).

DIR-E-I-L
This parameter specifies the use of the directory file if the DIRECTORY operand was not specified in the EXPORT, IMPORT, or LIST statements.

DIR-R-Q
This parameter specifies the use of the directory file if the DIRECTORY operand was not specified in the RESTORE or INQUIRE statements.

DIR-S-PO-PU
This parameter specifies the use of the directory file if the DIRECTORY operand was not specified in the SAVE, POOL or PURGE operands.

DIRSAVE
This parameter specifies whether the current directory file should be saved to the volume in the current save if the operand of the same name was not specified in the SAVE or EXPORT operands.

DUPLICATE
This parameter defines whether a file or job variable that has been specified for saving several times should actually be saved several times if the operand of the same name was not specified in the SAVE or EXPORT statements.

ERLIST
This parameter determines whether affected files should be listed in the event of an error even if the `LIST=NONE` operand was specified in the statement.

FGERASE
This parameter defines the behavior of an incremental save of file generations. The operand `FGERASE=YES` saves only the changed generations of a file generation group in a partial save (`CHANGED=YES`). If a RESTORE run of a specified save version is later performed with the `REPLACE=ALLP` operand, all unsaved file generations are lost.
The operand `FGERASE=NO` saves all generations in a file generation group if at least one generation needs to be saved, alternatively, no generations are saved.
If a RESTORE run is later performed with the `REPLACE=ALLP` operand, all generations are recovered without loss.

LIST-INQUIRE
This parameter determines the output medium for the report of the ARCHIVE run if the `LIST` operand was not specified in the INQUIRE or SHOW-DEFAULTS statements. The presetting `TASK-TYPE-DEPENDING` reflects the different default values of the statement in interactive and batch mode.

LIST-OTHERS
This parameter specifies the output medium for the report of the ARCHIVE run if the `LIST` operand was not specified in the statements (with the exception of the INQUIRE and SHOW-DEFAULTS statements).

NOW
This parameter determines when a statement is carried out if the operand of the same name was not specified in the respective statement (SAVE etc.).

OLS
This parameter controls whether an online save of UDS databases (with open status) should be permitted.

OPERATOR
This parameter determines whether ARCHIVE processes ARCHIVE messages requiring an operator response using standard processing settings, without outputting the message to the operator terminal or whether ARCHIVE outputs these messages and awaits an operator response.

PRIMARY
This parameter specifies the primary allocation of the save file if the operand of the same name was not specified in the SAVE or EXPORT statements.

RESTART
This parameter determines whether check points that permit an interrupted run to be continued with the PROCESS statement are written in the checkpoint file.

RETPD
This parameter defines a retention period in days for a save version if the operand of the same name was not specified in the SAVE or EXPORT statement.

SAVE-ACL
This parameter determines whether the ACL entries (SECOS up to V3.0) are stored with the saved files if the operand of the same name was not specified in the SAVE statement.

SECONDARY
When saving to hard disk, this parameter specifies the secondary allocation of save files if the operand of the same name was not specified in the SAVE or EXPORT statement.

SHARE-EXPORT
This parameter determines whether the new volumes or save files are cataloged as “shareable” if the operand of the same name was not specified in the EXPORT statement.

SHARE-SAVE
This parameter determines whether the new volumes or save files are cataloged as “shareable” if the SHARE operand was not specified in the SAVE statement.

SHUPDW
This parameter defines the wait time in seconds in the event of a share update conflict on the directory file.

SNR
This parameter (Saved-Not-Restored) determines whether files and job variables that are not written back because they already existed are output in the ARCHIVE report when recovering with the IMPORT/RESTORE statements with the `REPLACE=NO` operand.

SPACE
This parameter specifies whether or not files to be replaced by files with the same names from the save run are to be deleted before restoration if the operand of the same name was not specified in the IMPORT or RESTORE statements.

STREAM
This parameter determines whether the input and output operations on tape devices are performed in streaming mode.

TARGET-ARCHIVE-VERSION
This parameter determines whether the catalog entry for exported files is converted into a format that can be read by older ARCHIVE versions (< V2.6B) if the operand of the same name was not specified in the EXPORT statement.

TIME
This determines the number of CPU seconds available to this process once it is started if the `NOW=NO` operand was specified (either explicitly or as an ARCHIVE parameter) in the respective statements (e.g. SAVE etc) without a time specification.

TYPE
This parameter determines which ARCHIVE processes are listed if the operand of the same name was not specified in the STATUS statement.

UNLOAD
This parameter determines whether magnetic tapes are unloaded after being processed. The mount parameters defined at system level can also affect the actual behavior.

UNUSED
This parameter defines a selection of files to be saved, on the basis of the date of the last read or write access if the operand of the same name is not specified in the SAVE or EXPORT statements.

VOLUME
This parameter determines the VSNs of the output volumes if the operand of the same name was not specified in the SAVE or EXPORT statements.

WRCHK
This determines whether a read-after-write check (reading the file every time new content is added) is conducted when writing to hard disk, both when recovering files and when saving save files to disk.

STATUS
Display the status of ARCHIVE processes

You can use the STATUS statement to check the status of an ARCHIVE process. A process may either be active (A) or it may be in the queue file `ARCHIVE.LATER` (L) if the operand NOW=NO was specified. It may also already have been partly processed and interrupted by a system crash.

Nonprivileged users can only gain access to information about their own ARCHIVE processes with the STATUS statement. Information about all ARCHIVE processes is only output to users with the TSOS privilege.

You can start processes in the file `ARCHIVE.LATER` or continue interrupted processes with the PROCESS statement.

The STATUS information is always output to SYSOUT.

STA[TUS]
TYPE = LATER / ACTIVE [,USER = userid]

TYPE =
This determines which ARCHIVE processes are listed.

TYPE = LATER
All ARCHIVE processes for which NOW=NO has been specified and which have not yet been started are listed.

TYPE = ACTIVE
All ARCHIVE processes which have already been started but not yet terminated are listed. For processes currently being processed the TSN is also output. Processes which have been interrupted can either be continued by means of the PROCESS statement or deleted by means of the DELETE statement.

USER = userid
This only applies to systems support.
If ARCHIVE is called using the TSOS privilege and this operand is not specified, the ARCHIVE processes of all users are listed.
If systems support specifies this operand, information will be output about the ARCHIVE processes of the user IDs entered.

Example

See page 32.

7 Application examples

The tables below are intended to provide a quick overview of the application examples described in this chapter.

Simple applications

Example	ARCHIVE run	Directory file	Input from	Output to
1	Saving files	with	any volume	magnetic tape / MTC
2	Extending saves	with	any volume	magnetic tape / MTC
3	Restoration with the directory file	with	magnetic tape / MTC	original volume
4	Restoration without the directory file	without	magnetic tape / MTC	PUBLIC
5	Restoration to other pubsets	with	magnetic tape / MTC	any volume
6	Exporting files	without	any volume	magnetic tape / MTC
7	Importing files	without	magnetic tape / MTC	original volume
8	Duplication of save data	with	magnetic tape / MTC	magnetic tape / MTC
9	Duplication of save data	without	magnetic tape / MTC	magnetic tape / MTC

Complex application examples

Example	ARCHIVE run	Directory file	Input from	Output to
10	EXPORT of files with resetting of the assignment and the protection attributes	without	any volume	magnetic tape / MTC
11	IMPORT of files whose assignment and protection attributes have been reset	without	magnetic tape / MTC	original volume
12	SAVE (parallel processing)	with	any volume	magnetic tape / MTC
13	Merging of save data from different volumes	with	any volume	magnetic tape / MTC
14	Saving individual file generations	with	any volume	magnetic tape / MTC

Examples of complete system saves

Example	ARCHIVE run	Directory file	Input from	Output to
15	Full save	with	PUBLIC	magnetic tape / MTC
16	Incremental save	with existing directory file	PUBLIC	magnetic tape / MTC
17	Restoration from system backup	$TSOS. directory	magnetic tape / MTC	original volume
18	Updating the system backup	with	magnetic tape / MTC	magnetic tape / MTC

7.1 Simple applications

7.1.1 Example 1: Saving files

Saving files with a directory file from a public volume to magnetic tape or MT cartridge.

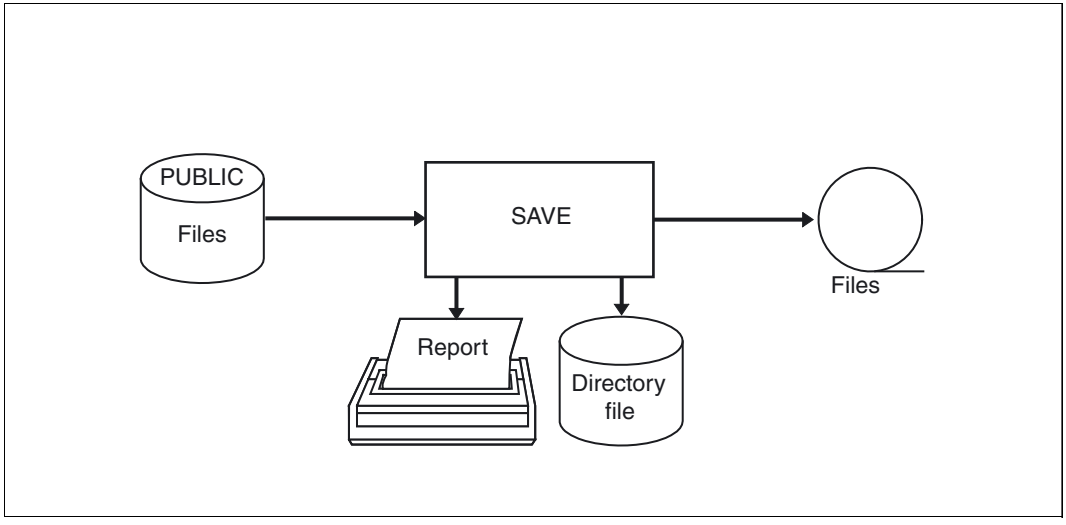


Figure 6: Saving files

```
/START-ARCHIVE ──────────────────────────────────────────────────────────── (1)
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY ──────────────────────────────────────────────── (2)
*FILES NAME=FILE. ──────────────────────────────────────────────────────── (3)
*SAVE DIRECTORY=DIR.SAVE.FILE,NEW,VOLUME=QE0829,
      DEVICE=TAPE-C4,LIST=BOTH ─────────────────────────────────────────── (4)
*         S A V E   C O M M A N D   L I S T I N G *** ──────────────────── (5)


PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = NO
STREAM  = YES

FILES NAME=FILE.
SAVE DIRECTORY=DIR.SAVE.FILE,NEW,VOLUME=QE0829,DEVICE=TAPE-C4,LIST=BOTH
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100817.142326',
           VERSION '9.0A' ──────────────────────────────────────────────── (6)
%  ARC0033 ARCHIVE SUBTASK TSN 'OGKZ' GENERATED ──────────────────────── (7)
%  MARM121 MAREN CATALOG ENTRY 'QE0829'/'0001' MODIFIED ──────────────── (8)
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '3646' PAM PAGES FOR '6' FILES
           AND '0' JVS IN '5' SECONDS ──────────────────────────────────── (9)
 ***         S A V E   A R C H I V E   R E P O R T ***        10-08-17        PAGE    1  (10)
                    SAVE VERSION IDENTIFIER - S.100817.142326 - ON VOLUME TYPE   TAPE-C4
SUBSAVE
 NUMBER       VSNS

    0       QE0829
      ***         S A V E   A R C H I V E   R E P O R T ***         10-08-17         PAGE    2
                         SAVE VERSION IDENTIFIER - S.100817.142326 - ON VOLUME TYPE   TAPE-C4
                               ***   USER - TSOS       ***
FILE/JOB VARIABLE NAME                    LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                                  VERS       SIZE  TYPE    VSN  TYP SAVE  VSN(S)

FILE.1                               1         64  FULL  BUR5.2  D     0  QE0829
FILE.2                               1        234  FULL  BUR5.0  D     0  QE0829
FILE.3                               1         64  FULL  BUR5.1  D     0  QE0829
FILE.4                               1        234  FULL  BUR5.2  D     0  QE0829
FILE.5                               1       1500  FULL  BUR5.2  D     0  QE0829
FILE.6                               1       1500  FULL  BUR5.0  D     0  QE0829
   ***     E N D   O F   A R C H I V E   R E P O R T ***         10-08-17         PAGE    3

%  ARC0003 ARCHIVE STATEMENT COMPLETED ──────────────────────────────────── (11)
*END ──────────────────────────────────────────────────────────────────── (12)
%  ARC0009 ARCHIVE TERMINATED ──────────────────────────────────────────── (13)
```

(1) ARCHIVE is called.

(2) ARCHIVE is loaded and ready.

(3) The NAME operand in the FILES statements specifies the files to be processed by ARCHIVE.

(4) The SAVE statement defines a save to MT cartridge (German abbreviation: MBK):

- DIRECTORY=DIR.SAVE.FILE,NEW
  ARCHIVE should write the information about the save run to the directory file. The `NEW` operand must be specified in order that the directory file is created. `NEW` must also be specified if the directory file has already been cataloged but does not have an entry in ARCHIVE.
- VOLUME=QE0829
  The MT cartridge with the specified VSN is defined as the save volume and added to the volume pool.
- DEVICE=TAPE-C4
  This operand specifies the device type for the MT cartridge.
- LIST=BOTH
  ARCHIVE is to output the report of the run to SYSLST and SYSOUT.

(5) For verification purposes ARCHIVE outputs a list containing the operand values valid for the run and the sequences of statements input.

(6) ARCHIVE confirms that the statements input are syntactically correct. The ARCHIVE run is assigned a unique ARCHIVE sequence number.

(7) ARCHIVE initiates a subtask for input/output with its own TSN.

(8) A MAREN subsystem message.

(9) ARCHIVE reports the total number of transferred files.

(10) Report of the save run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

(11) ARCHIVE reports that the save run was terminated without errors.

(12) Termination of ARCHIVE is requested.

(13) The ARCHIVE run is terminated.

7.1.2 Example 2: Extending saves

Extending an existing save version with a directory file on magnetic tape or MT cartridge.

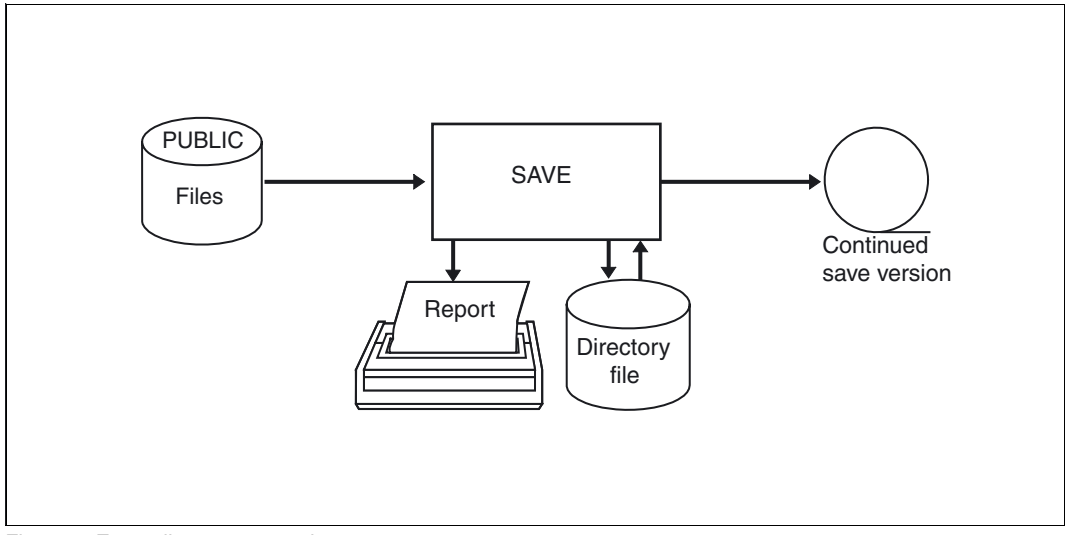


Figure 7: Expanding a save version

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES NAME=(FILE.7,FILE.8) ------------------------------------------------  (1)
*SAVE DIR=DIR.SAVE.FILE,CONTINUE=S.100817.142326,LIST=BOTH ----------------  (2)
*         S A V E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = NO
STREAM  = YES
```

```
FILES NAME=(FILE.7,FILE.8)
SAVE DIRECTORY=DIR.SAVE.FILE,CONTINUE=S.100817.142326,LIST=BOTH
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100817.142853',
           VERSION '9.0A'
%  MARM121 MAREN CATALOG ENTRY 'QE0829'/'0001' MODIFIED
%  ARC0033 ARCHIVE SUBTASK TSN 'OGK7' GENERATED
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '387' PAM PAGES FOR '2' FILES
           AND '0' JVS IN '1' SECONDS
 ***         S A V E   A R C H I V E   R E P O R T  ***          10-08-17        PAGE    1    (3)
                    SAVE VERSION IDENTIFIER - S.100817.142326 - ON VOLUME TYPE   TAPE-C4
SUBSAVE
 NUMBER       VSNS

    0        QE0829
      ***          S A V E   A R C H I V E   R E P O R T  ***          10-08-17          PAGE    2
                   SAVE VERSION IDENTIFIER - S.100817.142326 - ON VOLUME TYPE   TAPE-C4
                        ***   USER - TSOS       ***
FILE/JOB VARIABLE NAME                  LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                                VERS       SIZE  TYPE    VSN  TYP SAVE  VSN(S)

FILE.7                             1        192  FULL  BUR5.1  D     0  QE0829
FILE.8                             1        192  FULL  BUR5.1  D     0  QE0829
   ***      E N D   O F   A R C H I V E   R E P O R T  ***          10-08-17          PAGE    3

%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) The NAME operand in the FILES statement specifies the files to be processed by ARCHIVE.

(2) The SAVE statement instructs ARCHIVE to save to MT cartridge.

- DIRECTORY=DIR.SAVE.FILE
  ARCHIVE is to write the information about the save run to the directory file. ARCHIVE takes the VSN of the volume to be continued from the directory file. The TAPES/VOLUME operand need only be specified if continuation cartridges are to be defined.
- CONTINUE=S.100817.142326
  ARCHIVE is to extend the save version created in the previous example to include the files specified in the FILES statement.
- LIST=BOTH
  ARCHIVE is to output the run report to SYSLST and SYSOUT.

(3) Report of the restoration run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

7.1.3 Example 3: Restoration with the directory file

Writing files with a directory file from magnetic tape or MT cartridge back to the original volume.

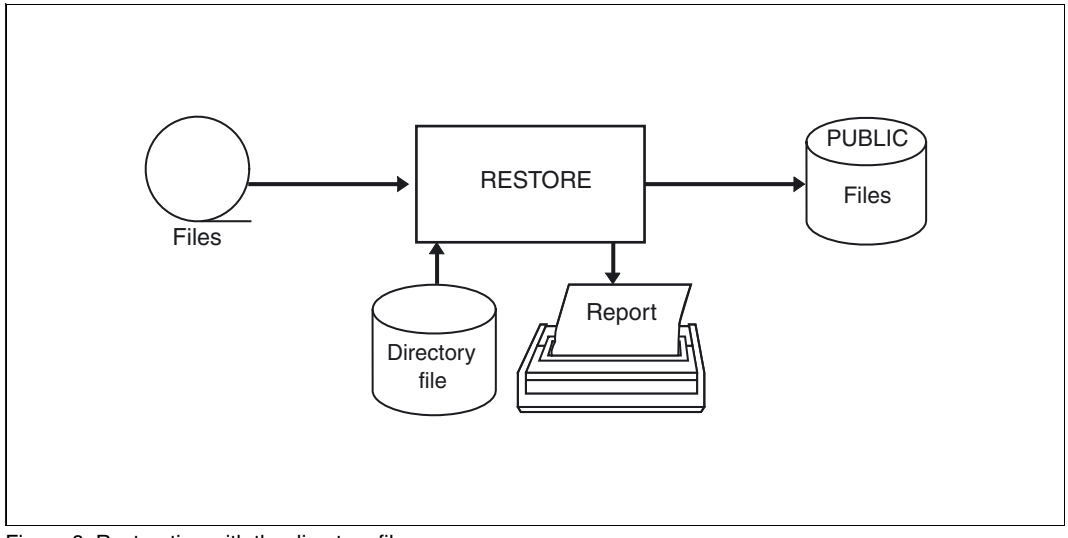


Figure 8: Restoration with the directory file

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES NAME=FILE. ------------------------------------------------------------  (1)
*RESTORE DIRECTORY=DIR.SAVE.FILE,LIST=SYSOUT ---------------------------------  (2)
*   R E S T O R E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
SNR     = YES
CATID   = NO
STREAM  = YES
```

```
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100817.143156',
          VERSION '9.0A'
% ARC0033 ARCHIVE SUBTASK TSN 'OGK9' GENERATED
% ARC0815 SUBTASK '0' HAS TRANSFERRED '4019' PAM PAGES FOR '8' FILES
          AND '0' JVS IN '5' SECONDS
 ***    R E S T O R E   A R C H I V E   R E P O R T  ***          10-08-17        PAGE    1    (3)
                      SAVE VERSION IDENTIFIER - S.100817.142326
SUBSAVE
 NUMBER       VSNS

    0        QE0829
      ***    R E S T O R E   A R C H I V E   R E P O R T  ***          10-08-17          PAGE    2
                              ***   USER - TSOS       ***
FILE/JOB VARIABLE NAME                    LASTPG/    SAVE VERSION  SAVE  INPUT    SUB OUTPUT
                                  VERS       SIZE     IDENTIFIER   TYPE   VSN    SAVE DISK(S)

FILE.1                               1         64  100817.142326  FULL  QE0829    0  BUR5.3
FILE.2                               1        234  100817.142326  FULL  QE0829    0  BUR5.3
FILE.3                               1         64  100817.142326  FULL  QE0829    0  BUR5.3
FILE.4                               1        234  100817.142326  FULL  QE0829    0  BUR5.3
FILE.5                               1       1500  100817.142326  FULL  QE0829    0  BUR5.3
FILE.6                               1       1500  100817.142326  FULL  QE0829    0  BUR5.3
FILE.7                               1        192  100817.142326  FULL  QE0829    0  BUR5.3
FILE.8                               1        192  100817.142326  FULL  QE0829    0  BUR5.3
   ***      E N D   O F   A R C H I V E   R E P O R T  ***          10-08-17          PAGE    3

% ARC0003 ARCHIVE STATEMENT COMPLETED
*END
% ARC0009 ARCHIVE TERMINATED
```

(1) The NAME operand in the FILES statements specifies the files to be processed by ARCHIVE.

(2) The RESTORE statement instructs ARCHIVE to restore saved files.

- DIRECTORY=DIR.FILE.SAVE
  ARCHIVE is to draw the necessary information for restoring the specified files from the directory file. In this case, the files will be restored to a public volume (from where they were saved) since no TO operand is specified in the FILES statement.
  ARCHIVE takes the VSN of the save volume from the directory file. The FROM operand can therefore be omitted.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT.
  By default the report is output to SYSLST.

(3) Report of the restoration run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

7.1.4 Example 4: Restoration without the directory file

Writing files that were saved from a private disk to magnetic tape or MT cartridge without a directory file back to a public volume.

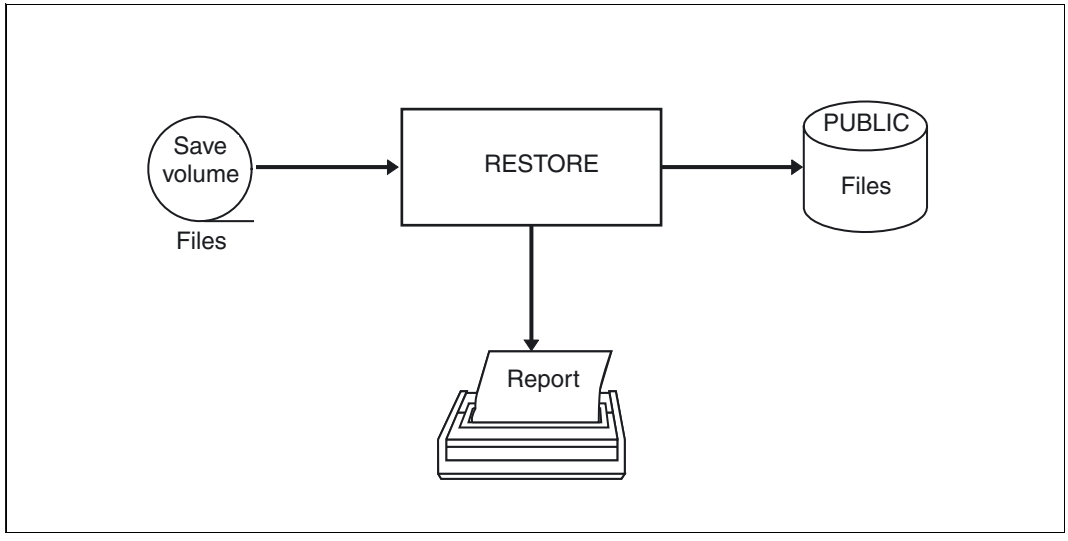


Figure 9: Restoration without the directory file

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES NAME=(PRIV.1,PRIV.2,PRIV.3),TO=PUBLIC ————————————————————————  (1)
*RESTORE DIRECTORY=NONE,FROM=(QE0829),LIST=SYSOUT ————————————————————  (2)
*   R E S T O R E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
SNR     = YES
CATID   = NO
STREAM  = YES
```

```
   FILES NAME=(PRIV.1,PRIV.2,PRIV.3),TO=PUBLIC
   RESTORE DIRECTORY=NONE,FROM=(QE0829),LIST=SYSOUT
   %  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100817.144703',
              VERSION '9.0A'
   %  ARC0033 ARCHIVE SUBTASK TSN 'OGLC' GENERATED
    ***    R E S T O R E   A R C H I V E   R E P O R T  ***         10-08-17        PAGE    1    (3)
                        SAVE VERSION IDENTIFIER - S.100817.144321
SUBSAVE
 NUMBER       VSNS

    0        QE0829
      ***    R E S T O R E   A R C H I V E   R E P O R T  ***         10-08-17         PAGE    2
                               ***   USER - TSOS       ***
FILE/JOB VARIABLE NAME                    LASTPG/   SAVE VERSION  SAVE  INPUT   SUB OUTPUT
                                  VERS       SIZE    IDENTIFIER   TYPE   VSN   SAVE DISK(S)

PRIV.1                               1        234  100817.144321  FULL  QE0829   0  BUR5.3
PRIV.2                               1        192  100817.144321  FULL  QE0829   0  BUR5.3
PRIV.3                               1        234  100817.144321  FULL  QE0829   0  BUR5.3
      ***      E N D   O F   A R C H I V E   R E P O R T  ***         10-08-17         PAGE    3

   %  ARC0003 ARCHIVE STATEMENT COMPLETED
   *END
   %  ARC0009 ARCHIVE TERMINATED
```

(1) The NAME operand in the FILES statements specifies the files to be processed in ARCHIVE run.

- TO=PUBLIC
  This operand is necessary since the files were saved from a private volume and would otherwise be written back there by default.

(2) The RESTORE statement writes the files back that were previously saved to MT cartridge in a SAVE run.

- DIRECTORY=NONE
  ARCHIVE is to restore the files from the save tape without any information from the directory file.
- FROM=(QE0829)
  The FROM operand specifies that the files are to be read from the magnetic tape with the specified VSN. The FROM operand must be specified when writing back without the directory file.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT.
  By default, the report is output to SYSLST.

(3) Report of the restoration run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

7.1.5 Example 5: Restoration to other pubsets

Writing files and job variables with a directory file from a save tape or MT cartridge to pubsets other than the originals.

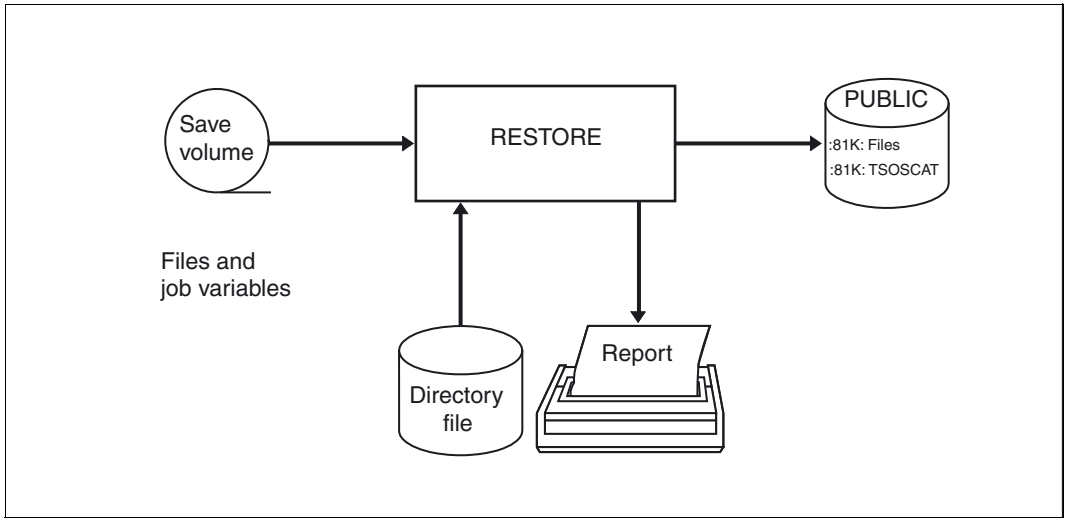


Figure 10: Restoration to other pubsets

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*PARAM CATID=YES ------------------------------------------------------------ (1)
%  ARC0017 PARAM STATEMENT ACCEPTED
*FILES NAME=(FILE.,RENAME=:81K:FILE.)---------------------------------------- (2)
*JOBVAR NAME=(JV1.,RENAME=:81K:JV1.)----------------------------------------- (3)
*RESTORE DIRECTORY=DIR.SAVE.FILE,LIST=SYSOUT -------------------------------- (4)
*   R E S T O R E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
SNR     = YES
CATID   = YES
STREAM  = YES

FILES NAME=(FILE.,RENAME=:81K:FILE.)
JOBVAR NAME=(JV1.,RENAME=:81K:JV1.)
RESTORE DIRECTORY=DIR.SAVE.FILE,LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100817.150047',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'OGLH' GENERATED
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '616' PAM PAGES FOR '4' FILES
           AND '3' JVS IN '0' SECONDS
 ***   R E S T O R E   A R C H I V E   R E P O R T ***        10-08-17       PAGE    1   (5)
                        SAVE VERSION IDENTIFIER - S.100817.145629
SUBSAVE
 NUMBER       VSNS

    0       QE0829
      ***   R E S T O R E   A R C H I V E   R E P O R T ***        10-08-17        PAGE    2
          ***   CATALOG - 81K       USER - TSOS       ***
FILE/JOB VARIABLE NAME                    LASTPG/   SAVE VERSION  SAVE  INPUT   SUB OUTPUT
                                  VERS       SIZE    IDENTIFIER   TYPE   VSN   SAVE DISK(S)

FILE.1                                       1         64  100817.145629  FULL  QE0829   0  81K.02
   RENAMED FROM :BUR5:$TSOS    .FILE.1
FILE.2                                       1        234  100817.145629  FULL  QE0829   0  81K.02
   RENAMED FROM :BUR5:$TSOS    .FILE.2
FILE.3                                       1         64  100817.145629  FULL  QE0829   0  81K.02
   RENAMED FROM :BUR5:$TSOS    .FILE.3
FILE.4                                       1        234  100817.145629  FULL  QE0829   0  81K.02
   RENAMED FROM :BUR5:$TSOS    .FILE.4
JV1.01                                       0          5  100817.145629    JV  QE0829   0
   RENAMED FROM :BUR5:$TSOS    .JV1.01
JV1.02                                       0          5  100817.145629    JV  QE0829   0
   RENAMED FROM :BUR5:$TSOS    .JV1.02
JV1.03                                       0          5  100817.145629    JV  QE0829   0
   RENAMED FROM :BUR5:$TSOS    .JV1.03
```

```
   ***     E N D   O F   A R C H I V E   R E P O R T  ***        10-08-17        PAGE    3

%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) This statement specifies that the catalog ID is to be used.

(2) The NAME operand in the FILES statement specifies the files to be processed by ARCHIVE. The RENAME operand defines a different file name when reading in and a different pubset as the destination.

(3) The NAME operand in the JOBVAR statement specifies the job variables that are to be processed by ARCHIVE. The RENAME operand defines a different file name when reading in and a different pubset as the destination.

(4) The RESTORE statement instructs ARCHIVE to restore previously saved files and job variables.

- DIRECTORY=DIR.SAVE.FILE
  ARCHIVE is to draw the information required for the restoration run from the directory file.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT.
  By default the report is output to SYSLST.

(5) Report of the restoration run. The processed files and job variables are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52f.

7.1.6 Example 6: Exporting files

Exporting files and job variables from your own ID and from a foreign user ID to magnetic tape or MT cartridge.

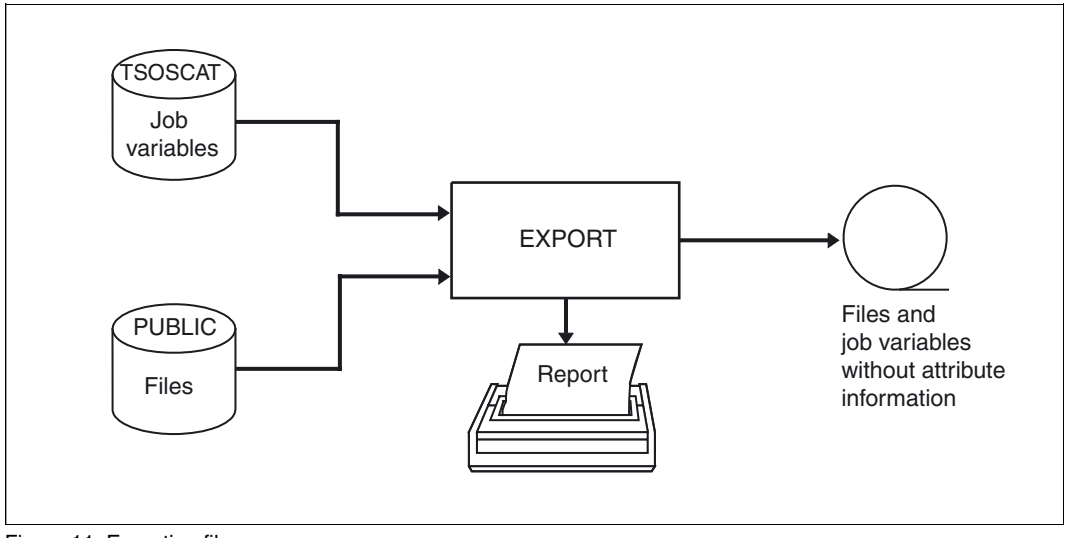


Figure 11: Exporting files

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES NAME=(FILE.,$USERID2.FILE.) ------------------------------------- (1)
*JOBVAR NAME=(JV.,$USERID2.JV.) ---------------------------------------- (2)
*EXPORT VOLUME=QE0830,DEVICE=TAPE-C4,LIST=SYSOUT,ATTRIBUTES=RESET ------ (3)
*     E X P O R T   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = NO
STREAM  = YES

FILES NAME=(FILE.,$USERID2.FILE.)
JOBVAR NAME=(JV.,$USERID2.JV.)
EXPORT VOLUME=QE0830,DEVICE=TAPE-C4,LIST=SYSOUT,ATTRIBUTES=RESET
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100824.132029',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN '0A4U' GENERATED
%  MARM127 VOLUME 'QE0830' RESERVED FOR USER ID 'USERID1'
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '2222' PAM PAGES FOR '6' FILES
           AND '4' JVS IN '4' SECONDS
 ***     E X P O R T   A R C H I V E   R E P O R T ***         10-08-24        PAGE    1   (4)
                      SAVE VERSION IDENTIFIER - S.100824.132029 - ON VOLUME TYPE   TAPE-C4
SUBSAVE
 NUMBER       VSNS

    0       QE0830
   ***     E X P O R T   A R C H I V E   R E P O R T  ***         10-08-24         PAGE    2
                      SAVE VERSION IDENTIFIER - S.100824.132029 - ON VOLUME TYPE   TAPE-C4
                           ***   USER - USERID1    ***
FILE/JOB VARIABLE NAME                  LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                              VERS        SIZE  TYPE    VSN  TYP SAVE  VSN(S)

FILE.01                          1         114  FULL  BUR5.1  D     0  QE0830
FILE.02                          1         156  FULL  BUR5.1  D     0  QE0830
FILE.03                          1         800  FULL  BUR5.2  D     0  QE0830
JV.01                            0           3    JV          C     0  QE0830
JV.02                            0           8    JV          C     0  QE0830
```

```
          ***     E X P O R T   A R C H I V E   R E P O R T  ***          10-08-24          PAGE    3
                                SAVE VERSION IDENTIFIER - S.100824.132029 - ON VOLUME TYPE   TAPE-C4
                                     ***   USER - USERID2    ***

        FILE/JOB VARIABLE NAME                    LASTPG/  SAVE    INPUT DEV  SUB  OUTPUT
                                          VERS       SIZE  TYPE     VSN  TYP SAVE  VSN(S)

FILE.01                                      1        114  FULL  BUR5.3  D     0  QE0830
FILE.02                                      1        156  FULL  BUR5.1  D     0  QE0830
FILE.03                                      1        800  FULL  BUR5.2  D     0  QE0830
JV.03                                        0         11    JV          C     0  QE0830
JV.04                                        0         15    JV          C     0  QE0830
          ***     E N D   O F   A R C H I V E   R E P O R T  ***          10-08-24          PAGE    4
```

```
% ARC0003 ARCHIVE STATEMENT COMPLETED
*END
% ARC0009 ARCHIVE TERMINATED
```

(1) The NAME operand in the FILES statement identifies the files under your own ID and those under the user ID `USERID2` that are to be processed by ARCHIVE.

(2) The NAME operand in the JOBVAR statement identifies the job variables under your own ID and those under the user ID `USERID2` that are to be processed be ARCHIVE.

(3) The EXPORT statement defines a save to MT cartridge that can be read in on a different BS2000/OSD system.

- VOLUME=QE0830
  This operand specifies the VSN of the save volume.
- DEVICE=TAPE-C4
  This operand specifies the device type for the MT cartridge.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT.
  By default the report is output to SYSLST.

(4) Report of the save run.
The processed files belonging to your own ID and those belonging to the foreign user ID `USERID2` are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

7.1.7 Example 7: Importing files

Reading in files and job variables belonging to your own ID and a foreign user ID from magnetic tape or MT cartridge.

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES NAME=(FILE.,$USERID2.FILE.,RENAME=ID2.FILE.) ————————————————————  (1)
*JOBVAR NAME=(JV.,$USERID2.JV.,RENAME=ID2.JV.) ——————————————————————————  (2)
*IMPORT FROM=(QE0830),DEVICE=TAPE-C4,LIST=SYSOUT ————————————————————————  (3)
*     I M P O R T   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
SNR     = YES
CATID   = NO
STREAM  = YES

FILES NAME=(FILE.,$USERID2.FILE.,RENAME=ID2.FILE.)
JOBVAR NAME=(JV.,$USERID2.JV.,RENAME=ID2.JV.)
IMPORT FROM=(QE0830),DEVICE=TAPE-C4,LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100824.132654',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN '0A4W' GENERATED
 ***     I M P O R T   A R C H I V E   R E P O R T ***         10-08-24        PAGE    1    (4)
                        SAVE VERSION IDENTIFIER - S.100824.132029
        SUBSAVE
         NUMBER       VSNS

            0        QE0830
              ***      I M P O R T   A R C H I V E   R E P O R T  ***         10-08-24         PAGE    2
                                     ***   USER - USERID1    ***
        FILE/JOB VARIABLE NAME                    LASTPG/    SAVE VERSION  SAVE  INPUT    SUB OUTPUT
                                       VERS         SIZE     IDENTIFIER   TYPE   VSN    SAVE DISK(S)

FILE.01                                   1          114   100824.132029  FULL  QE0830    0  BUR5.0
FILE.02                                   1          156   100824.132029  FULL  QE0830    0  BUR5.0
FILE.03                                   1          800   100824.132029  FULL  QE0830    0  BUR5.0
ID2.FILE.01                               1          114   100824.132029  FULL  QE0830    0  BUR5.0
   RENAMED FROM $USERID2 .FILE.01
ID2.FILE.02                               1          156   100824.132029  FULL  QE0830    0  BUR5.0
   RENAMED FROM $USERID2 .FILE.02
ID2.FILE.03                               1          800   100824.132029  FULL  QE0830    0  BUR5.0
   RENAMED FROM $USERID2 .FILE.03
```

```
ID2.JV.03                                  0        11  100824.132029    JV  QE0830   0
   RENAMED FROM $USERID2 .JV.03
ID2.JV.04                                  0        15  100824.132029    JV  QE0830   0
   RENAMED FROM $USERID2 .JV.04
JV.01                                      0         3  100824.132029    JV  QE0830   0
JV.02                                      0         8  100824.132029    JV  QE0830   0
                ***     E N D   O F   A R C H I V E   R E P O R T  ***        10-08-24         PAGE    3

            %  ARC0003 ARCHIVE STATEMENT COMPLETED
            *END
            %  ARC0009 ARCHIVE TERMINATED
```

(1) The NAME operand in the FILES statement identifies the files that are to be processed by ARCHIVE; the NAME operand in the JOBVAR statement specifies the job variables. The RENAME operand is necessary since the files and job variables are to be read in from the user ID `USERID2` to the foreign user ID `USERID1`.

If the TO operand is not specified in the FILES statement, ARCHIVE writes all files in an IMPORT run to public volumes by default. This applies even if the files were saved from private volumes.

(2) The IMPORT statement reads in files and job variables that were previously saved in an EXPORT run. If the files are read in using a different BS2000/OSD system, the catalog entry in TSOSCAT is created for them.

– FROM=(QE0830)
ARCHIVE is to write back the files and job variables specified in the FILES and JOBVAR statement from the MT cartridge with the corresponding VSN. The FROM operand is mandatory for an IMPORT run.

If the EXPORT run was performed with a directory file and the directory file was saved (`DIRSAVE=YES`), then this must be read in first with a separate IMPORT run using `DIRECTORY=NONE`, if the save needs to be read in using a different BS2000/OSD system. The required files and job variables can then be read in in an IMPORT run using the directory file.

– DEVICE=TAPE-C4
This operand specifies the device type for the MT cartridge.

– LIST=SYSOUT
ARCHIVE is to output the report of the run to SYSOUT.
The report is output to SYSLST by default.

(3) Report of the IMPORT run. The processed files and job variables are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

7.1.8 Example 8: Duplication of save data (with DIR)

Copying save data from magnetic tape or MT cartridge with the directory file to a different type of tape.

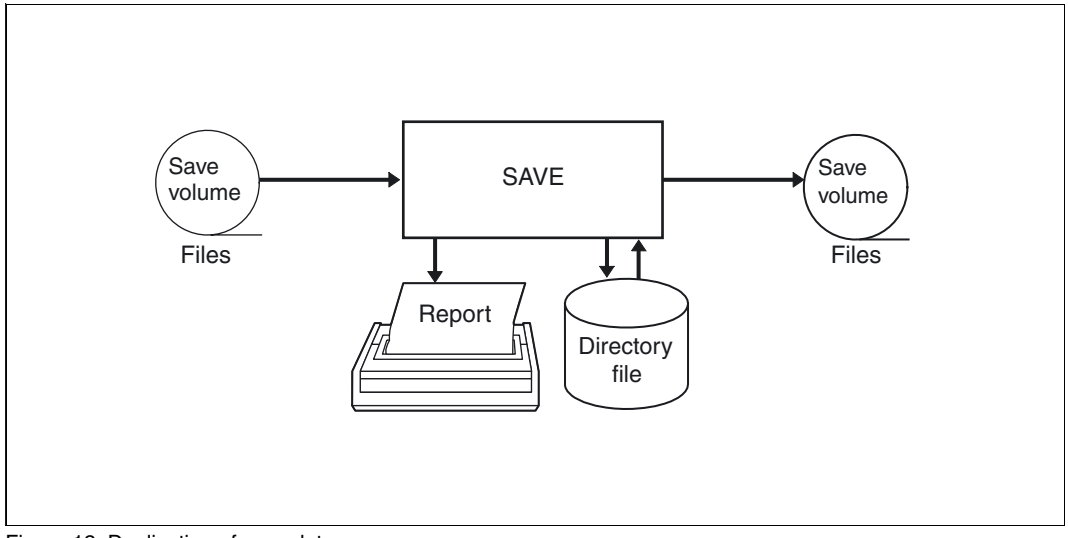


Figure 12: Duplication of save data

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES FROM=SV,(QE0829),DEVICE=TAPE-C4 ------------------------------------  (1)
*SAVE DIRECTORY=DIR.SAVEF,CHANGED=NO,VOLUME=00FFA0,DEVICE=TAPE-C6,
      LIST=SYSOUT ---------------------------------------------------------  (2)
*         S A V E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = NO
STREAM  = YES

FILES FROM=SV,(QE0829),DEVICE=TAPE-C4
SAVE DIRECTORY=DIR.SAVEF,CHANGED=NO,VOLUME=00FFA0,DEVICE=TAPE-C6,LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100819.101435',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN '0AI8' GENERATED
%  MARM127 VOLUME '00FFA0' RESERVED FOR USER ID 'TSOS'
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '2176' PAM PAGES FOR '6' FILES
           AND '0' JVS IN '16' SECONDS
 ***         S A V E   A R C H I V E   R E P O R T  ***          10-08-19        PAGE    1   (3)
                    SAVE VERSION IDENTIFIER - S.100819.101435 - ON VOLUME TYPE   TAPE-C6
SUBSAVE
 NUMBER       VSNS

    0        00FFA0
      ***          S A V E   A R C H I V E   R E P O R T  ***          10-08-19          PAGE    2
                   SAVE VERSION IDENTIFIER - S.100819.101435 - ON VOLUME TYPE   TAPE-C6
                        ***   USER - TSOS       ***
FILE/JOB VARIABLE NAME                  LASTPG/  SAVE    INPUT DEV  SUB  OUTPUT
                              VERS        SIZE  TYPE     VSN  TYP SAVE  VSN(S)

FILE.01                          1         114  FULL  QE0829  T     0  00FFA0
FILE.02                          1         156  FULL  QE0829  T     0  00FFA0
FILE.03                          1         800  FULL  QE0829  T     0  00FFA0
FILE.04                          1         114  FULL  QE0829  T     0  00FFA0
FILE.05                          1         156  FULL  QE0829  T     0  00FFA0
FILE.06                          1         800  FULL  QE0829  T     0  00FFA0
   ***      E N D   O F   A R C H I V E   R E P O R T  ***          10-08-19          PAGE    3


%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) ARCHIVE is to process the files from the source cartridge.

(2) The SAVE statement specifies a save to a different MT cartridge.

- DIRECTORY=DIR.SAVEF
  ARCHIVE is not to write information on the save run to a directory file.
- CHANGED=NO
  All files are always saved in full.
- VOLUME=00FFA0
  ARCHIVE is to save the files specified in the FILES statement to the MT cartridge with the corresponding VSN.
- DEVICE=TAPE-C6
  This operand specifies the device type of the output cartridge.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT. By default the report is output to SYSLST.

(3) Report of the save run. The processed files and job variables are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

7.1.9 Example 9: Duplication of save data (without DIR)

Copying save data from magnetic tape or MT cartridge without the directory file to a different type of tape.

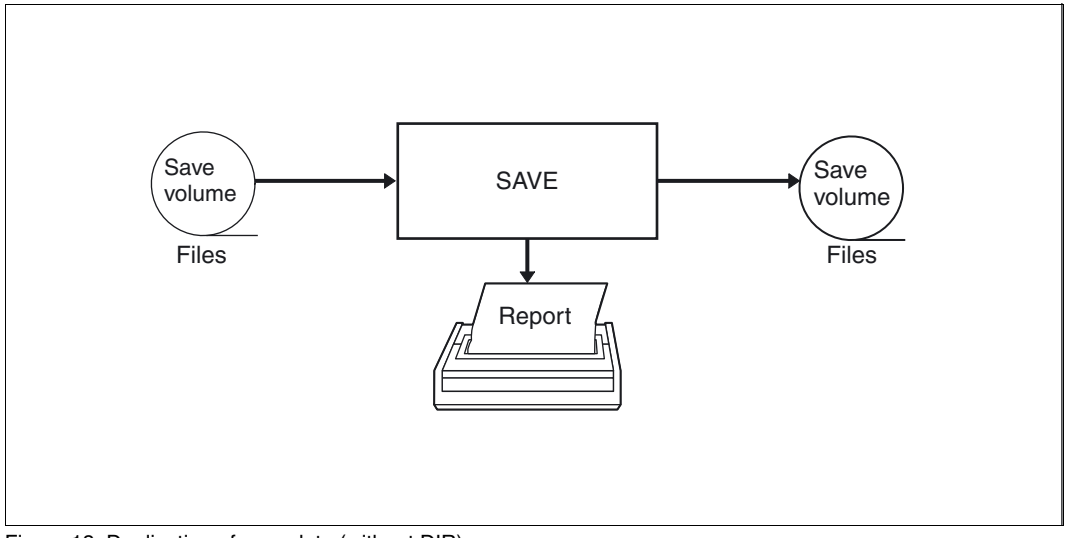


Figure 13: Duplication of save data (without DIR)

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES FROM=S.100819.081024,(QE0829),DEVICE=TAPE-C4 ---------------------  (1)
*SAVE DIRECTORY=NONE,CHANGED=NO,VOLUME=045DA9,DEVICE=TAPE-C6,
      LIST=SYSOUT ---------------------------------------------------------  (2)
*         S A V E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = NO
STREAM  = YES

FILES FROM=S.100819.081024,(QE0829),DEVICE=TAPE-C4
SAVE DIRECTORY=NONE,CHANGED=NO,VOLUME=045DA9,DEVICE=TAPE-C6,LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100819.090414',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN '0AI2' GENERATED
%  MARM127 VOLUME '045DA9' RESERVED FOR USER ID 'TSOS'
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '2176' PAM PAGES FOR '6' FILES
           AND '0' JVS IN '16' SECONDS
 ***         S A V E   A R C H I V E   R E P O R T ***         10-08-19        PAGE    1   (3)
                    SAVE VERSION IDENTIFIER - S.100819.090414 - ON VOLUME TYPE   TAPE-C6
SUBSAVE
 NUMBER       VSNS

    0        045DA9
      ***         S A V E   A R C H I V E   R E P O R T ***         10-08-19         PAGE    2
                  SAVE VERSION IDENTIFIER - S.100819.090414 - ON VOLUME TYPE   TAPE-C6
                        ***   USER - TSOS       ***
FILE/JOB VARIABLE NAME                LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                            VERS        SIZE  TYPE    VSN  TYP SAVE  VSN(S)

FILE.01                        1         114  FULL  QE0829  T     0  045DA9
FILE.02                        1         156  FULL  QE0829  T     0  045DA9
FILE.03                        1         800  FULL  QE0829  T     0  045DA9
FILE.04                        1         114  FULL  QE0829  T     0  045DA9
FILE.05                        1         156  FULL  QE0829  T     0  045DA9
FILE.06                        1         800  FULL  QE0829  T     0  045DA9
   ***     E N D   O F   A R C H I V E   R E P O R T ***         10-08-19         PAGE    3


%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) ARCHIVE is to process the files from the specified source cartridge.

(2) The SAVE statement defines a save to another MT cartridge.

- DIRECTORY=NONE
  ARCHIVE is not to write any information about the save run in a directory file.
- CHANGED=NO
  All files are always saved in full.
- VOLUME=045DA9
  ARCHIVE is to save the files specified in the FILES statement to the MT cartridge with the corresponding VSN.
- DEVICE=TAPE-C6
  This operand specifies the device type for the output cartridge.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT.
  The report is output to SYSLST by default.

(3) Report of the save run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

7.2 Complex applications

7.2.1 Example 10: Exporting files

Transferring files and job variables on MT cartridge so that they can be processed without assignments (catalog ID and user ID) and without information about password and access attributes.

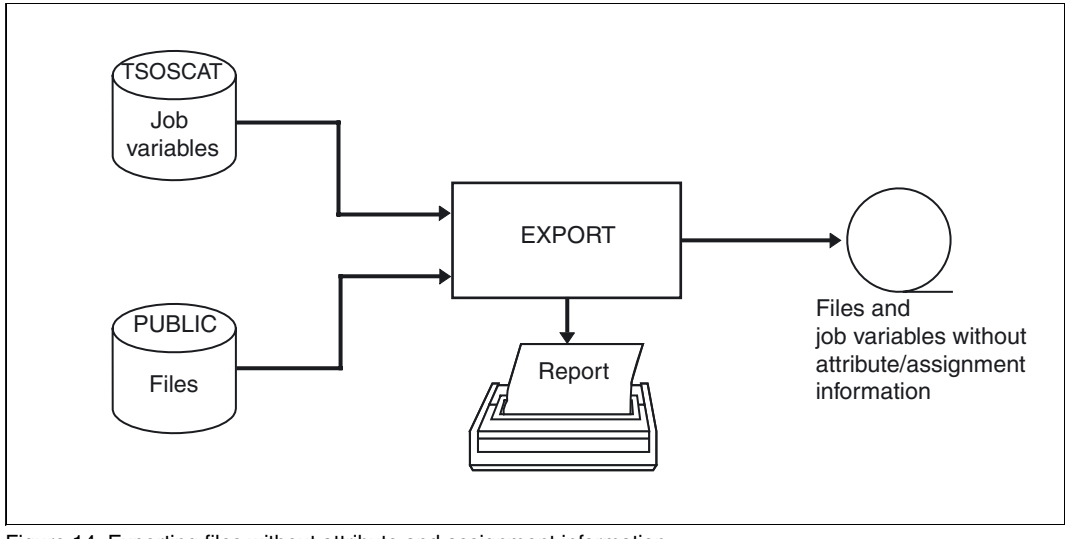


Figure 14: Exporting files without attribute and assignment information

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*PARAM CATID=YES ------------------------------------------------------------  (1)
%  ARC0017 PARAM STATEMENT ACCEPTED
*FILES NAME=FILE. -----------------------------------------------------------  (2)
*JOBVAR NAME=JV1. -----------------------------------------------------------  (3)
*EXPORT VOLUME=QE0798,DEVICE=TAPE-C4,LOCATION=RESET,ATTRIBUTES=RESET,
       LIST=SYSOUT -----------------------------------------------------------  (4)
*     E X P O R T   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = YES
STREAM  = YES

FILES NAME=FILE.
JOBVAR NAME=JV1.
EXPORT VOLUME=QE0798,DEVICE=TAPE-C4,LOCATION=RESET,ATTRIBUTES=RESET,
       LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100818.150912',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'ORJK' GENERATED
%  MARM121 MAREN CATALOG ENTRY 'QE0798'/'0001' MODIFIED
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '122' PAM PAGES FOR '3' FILES
           AND '2' JVS IN '1' SECONDS
 ***     E X P O R T   A R C H I V E   R E P O R T ***         10-08-18        PAGE    1  (5)
                    SAVE VERSION IDENTIFIER - S.100818.150912 - ON VOLUME TYPE   TAPE-C4
SUBSAVE
 NUMBER       VSNS

    0       QE0798
   ***     E X P O R T   A R C H I V E   R E P O R T ***         10-08-18         PAGE    2
                    SAVE VERSION IDENTIFIER - S.100818.150912 - ON VOLUME TYPE   TAPE-C4
                                    ***   LOCATION RESET   ***    ----------------------  (6)
FILE/JOB VARIABLE NAME                LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                            VERS        SIZE  TYPE    VSN  TYP SAVE  VSN(S)

FILE.1                         1          56  FULL  POST.2  D     0  QE0798
FILE.2                         1          25  FULL  POST.0  D     0  QE0798
FILE.3                         1          40  FULL  POST.1  D     0  QE0798
JV1.1                          0           5    JV          C     0  QE0798
JV1.2                          0          17    JV          C     0  QE0798
     ***     E N D   O F   A R C H I V E   R E P O R T ***         10-08-18         PAGE    3
```

```
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) This statement specifies that the catalog ID is to be used.

(2) The NAME operand of the FILES statement specifies the files to be processed by ARCHIVE.

(3) The NAME operand of the JOBVAR statement specifies the job variables to be processed by ARCHIVE.

(4) The EXPORT statement defines a save that can be read in using a different BS2000/OSD system.

- VOLUME=QE0798
  This operand specifies the VSN of the save volume.
- DEVICE=TAPE-C4
  This operand specifies the device type for the MT cartridge.
- LOCATION=RESET
  The information on the assignment of each exported file (catalog ID and user ID) is reset in the save file.
- ATTRIBUTES=RESET
  The password and access protection attributes are reset during the save run.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT.
  By default the report is output to SYSLST.

(5) Report of the EXPORT run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

(6) The output LOCATION RESET indicates that the assignment of the exported file (catalog ID and user ID) has been reset.

7.2.2 Example 11: Importing files

Importing files and job variables that were exported to magnetic tape or MT cartridge with reset password and access attributes and without catalog ID and user ID.

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES  ──────────────────────────────────────────────────────────────────  (1)
*JOBVAR
*IMPORT FROM=(QE0798),DEVICE=TAPE-C4,LIST=SYSOUT ──────────────────────  (2)
*     I M P O R T   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
SNR     = YES
CATID   = NO
STREAM  = YES

FILES
JOBVAR
IMPORT FROM=(QE0798),DEVICE=TAPE-C4,LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100818.152007',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'ORJN' GENERATED
 ***     I M P O R T   A R C H I V E   R E P O R T ***         10-08-18        PAGE    1  (3)
                                     ***   LOCATION RESET   ***     ──────────────────  (4)
          FILE/JOB VARIABLE NAME                 LASTPG/   SAVE VERSION  SAVE  INPUT   SUB OUTPUT
                                      VERS         SIZE    IDENTIFIER   TYPE   VSN   SAVE DISK(S)

FILE.1                                   1           56  100818.150912  FULL  QE0798   0  POST.1
FILE.2                                   1           25  100818.150912  FULL  QE0798   0  POST.1
FILE.3                                   1           40  100818.150912  FULL  QE0798   0  POST.1
JV1.1                                    0            5  100818.150912    JV  QE0798   0
JV1.2                                    0           17  100818.150912    JV  QE0798   0
   ***     E N D   O F   A R C H I V E   R E P O R T ***         10-08-18         PAGE    2

%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) Since no NAME operand was specified in the FILES statement or the JOBVAR statement, all files / job variables located on the MT cartridge are read in.

(2) With the IMPORT statement, the files and job variables that were previously saved in an EXPORT run are read in again. If the files are imported to another BS2000/OSD system, a catalog entry in TSOSCAT is also set up for this file.

– FROM=(QE0789)
ARCHIVE is to restore all files and job variables from the MT cartridge with the specified VSN.
Specification of the FROM operand is mandatory for an IMPORT run.

If a directory file was used in an EXPORT run and included in the save (`DIRSAVE=YES`), this directory file must first be restored in a separate IMPORT run with `DIRECTORY=NONE` if the save is to be imported to another BS2000/OSD system.
The required files and job variables can then be read in in an IMPORT run using the directory file.

– DEVICE=TAPE-C4
This operand specifies the device type for the MT cartridge.

– LIST=SYSOUT
ARCHIVE is to output the report of the run to SYSOUT.
By default the report is output to SYSLST.

(3) Report of the IMPORT run. The processed files and job variables are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

(4) The output LOCATION RESET indicates that the assignment information for the file (catalog ID and user ID) was reset at the time of the EXPORT run.

7.2.3 Example 12: Save, parallel processing

Saving files and job variables in parallel with a directory file.

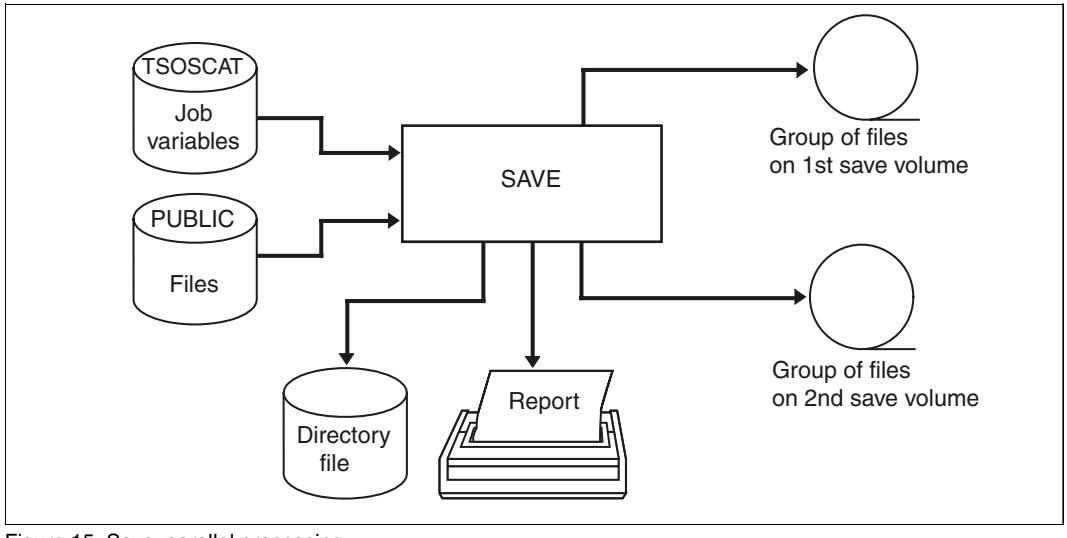


Figure 15: Save, parallel processing

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES NAME=(FILE.,OUTPUT.),NAME=(PROG.,TEST.) ————————————————————————  (1)
*JOBVAR NAME=JV1.,NAME=(JV2.,JV3.) ————————————————————————————————————  (2)
*SAVE DIRECTORY=DIR.SAVEP,NEW,TAPES=(QE0798,QE0799),DEVICE=TAPE-C4,
      DRIVES=2,LIST=SYSOUT ————————————————————————————————————————————  (3)
*         S A V E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = NO
STREAM  = YES
```

```
FILES NAME=(FILE.,OUTPUT.),NAME=(PROG.,TEST.)
JOBVAR NAME=JV1.,NAME=(JV2.,JV3.)
SAVE DIRECTORY=DIR.SAVEP,NEW,TAPES=(QE0798,QE0799),DEVICE=TAPE-C4,
     DRIVES=2,LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100818.160451',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'ORJ2' GENERATED ————————————————————————  (4)
%  ARC0033 ARCHIVE SUBTASK TSN 'ORJ3' GENERATED
%  MARM127 VOLUME 'QE0798' RESERVED FOR USER ID 'TSOS'
%  MARM127 VOLUME 'QE0799' RESERVED FOR USER ID 'TSOS'
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '241' PAM PAGES FOR '5' FILES
           AND '2' JVS IN '3' SECONDS
%  ARC0815 SUBTASK '1' HAS TRANSFERRED '1004' PAM PAGES FOR '5' FILES
           AND '3' JVS IN '3' SECONDS
 ***         S A V E   A R C H I V E   R E P O R T  ***          10-08-18        PAGE    1   (5)
                  SAVE VERSION IDENTIFIER - S.100818.160451 - ON VOLUME TYPE   TAPE-C4
SUBSAVE
 NUMBER       VSNS

   0        QE0798
   1        QE0799
     ***          S A V E   A R C H I V E   R E P O R T  ***          10-08-18          PAGE    2
                  SAVE VERSION IDENTIFIER - S.100818.160451 - ON VOLUME TYPE   TAPE-C4
                        ***   USER - TSOS       ***
FILE/JOB VARIABLE NAME                LASTPG/  SAVE    INPUT DEV  SUB  OUTPUT
                          VERS          SIZE  TYPE     VSN  TYP SAVE  VSN(S)

FILE.1                       1            56  FULL  POST.1  D     0  QE0798
FILE.2                       1            25  FULL  POST.1  D     0  QE0798
FILE.3                       1            40  FULL  POST.1  D     0  QE0798
JV1.1                        0             5    JV          C     0  QE0798
JV1.2                        0            17    JV          C     0  QE0798
JV2.1                        0             6    JV          C     1  QE0799
JV3.1                        0            13    JV          C     1  QE0799
JV3.2                        0            22    JV          C     1  QE0799
OUTPUT.1                     1            48  FULL  POST.2  D     0  QE0798
OUTPUT.2                     1            69  FULL  POST.0  D     0  QE0798
PROG.1                       1             4  FULL  POST.2  D     1  QE0799
PROG.2                       1            22  FULL  POST.0  D     1  QE0799
TEST.1                       1             9  FULL  POST.2  D     1  QE0799
TEST.2                       1             3  FULL  POST.0  D     1  QE0799
TEST.3                       1           928  FULL  POST.1  D     1  QE0799
   ***      E N D   O F   A R C H I V E   R E P O R T  ***          10-08-18          PAGE    3

%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) The NAME operand in the FILES statement specifies the files to be processed by ARCHIVE.

(2) The NAME operand in the JOBVAR statement specifies the job variables to be processed by ARCHIVE.

(3) The SAVE statement defines a save to tape.

- DIRECTORY=DIR.SAVEP,NEW
  ARCHIVE is to write the information on the save run to the directory file. The `NEW` operand must be specified here, as the directory file is created as a new file. `NEW` must also be specified if the file has already been cataloged but as yet contains no ARCHIVE entries.
- TAPES=(QE0798,QE0799)
  ARCHIVE is to save the files and job variables specified in the FILES and JOBVAR statements to the corresponding MT cartridge.
- DEVICE=TAPE-C4
  This operand specifies the device type for the MT cartridge.
- DRIVES=2
  Two parallel runs are defined for this run. This speeds up the save run.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT. By default the report is output to SYSLST.

(4) ARCHIVE initiates two subtasks for input/output with their own TSNs.

(5) Report of the save run. The processed files and job variables are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52. The report shows that ARCHIVE splits up the files specified in the NAME operand into packets for the respective parallel runs.

For runs with `PARAM CATID=YES`, ARCHIVE does not always split the files as specified in the FILES statement. Instead, ARCHIVE attempts to save the files and job variables of one catalog to one magnetic tape or MT cartridge.

7.2.4 Example 13: Merging of save data

Merging files and job variables from disk and tape or MT cartridge together on a different tape or MT cartridge.

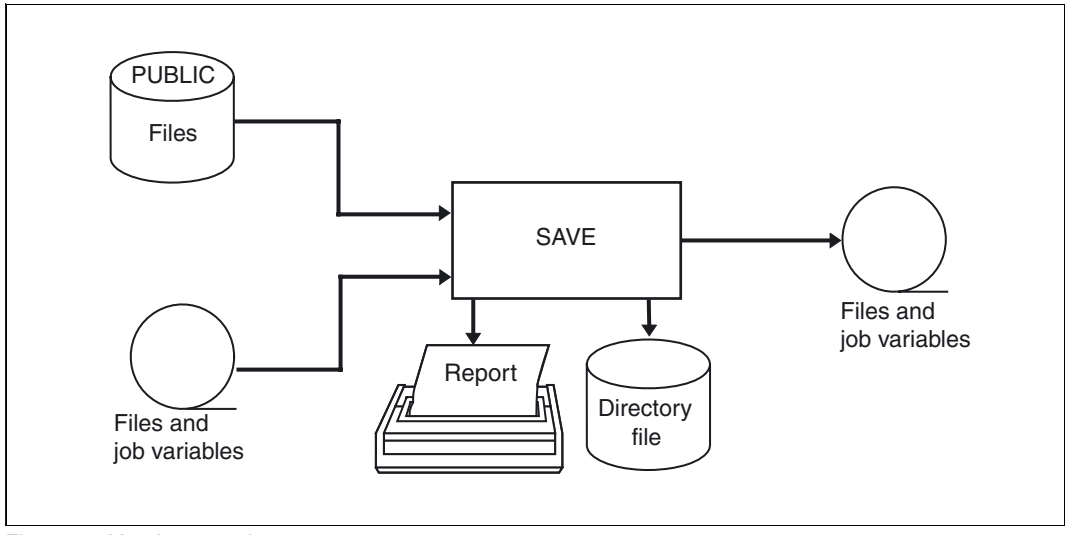


Figure 16: Merging save data

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*PARAM CATID=YES ------------------------------------------------------------ (1)
%  ARC0017 PARAM STATEMENT ACCEPTED
*FILES NAME=TEST. ----------------------------------------------------------- (2)
*FILES FROM=SV,(QE0798),DEVICE=TAPE-C4 -------------------------------------- (3)
*JOBVAR FROM=SV,(QE0798),DEVICE=TAPE-C4 ------------------------------------- (4)
*SAVE DIRECTORY=DIR.SAVES,VOLUME=QE0799,DEVICE=TAPE-C4,DUPLICATE=NO,
      LIST=SYSOUT ----------------------------------------------------------- (5)
*         S A V E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = YES
STREAM  = YES

FILES NAME=TEST.
FILES FROM=SV,(QE0798),DEVICE=TAPE-C4
JOBVAR FROM=SV,(QE0798),DEVICE=TAPE-C4
SAVE DIRECTORY=DIR.SAVES,VOLUME=QE0799,DEVICE=TAPE-C4,DUPLICATE=NO,
     LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100818.162948',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'ORKB' GENERATED
%  MARM127 VOLUME 'QE0799' RESERVED FOR USER ID 'TSOS'
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '1115' PAM PAGES FOR '7' FILES
           AND '2' JVS IN '1' SECONDS
 ***         S A V E   A R C H I V E   R E P O R T  ***         10-08-18        PAGE    1   (6)
                  SAVE VERSION IDENTIFIER - S.100818.162948 - ON VOLUME TYPE   TAPE-C4
SUBSAVE
 NUMBER       VSNS

   0        QE0799
     ***         S A V E   A R C H I V E   R E P O R T  ***         10-08-18         PAGE    2
                  SAVE VERSION IDENTIFIER - S.100818.162948 - ON VOLUME TYPE   TAPE-C4
          ***   CATALOG - POST      USER - TSOS       ***
FILE/JOB VARIABLE NAME                LASTPG/  SAVE    INPUT DEV  SUB  OUTPUT
                              VERS       SIZE  TYPE     VSN  TYP SAVE  VSN(S)

FILE.1                           1         56  FULL  QE0798  T     0  QE0799
FILE.2                           1         25  FULL  QE0798  T     0  QE0799
FILE.3                           1         40  FULL  QE0798  T     0  QE0799
JV1.1                            0          5    JV  QE0798  T     0  QE0799
```

```
JV1.2                                  0        17    JV  QE0798  T     0  QE0799
TEST.1                                 1         9  FULL  POST.2  D     0  QE0799
TEST.2                                 1         3  FULL  POST.0  D     0  QE0799
TEST.3                                 1       928  FULL  POST.1  D     0  QE0799
   ***     E N D   O F   A R C H I V E   R E P O R T  ***        10-08-18        PAGE    3

%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) This statement specifies that the catalog ID is to be used.

(2) The NAME operand of the FILES statement specifies the files to be processed by ARCHIVE.

(3) Since the NAME operand is not specified in the FILES statement, ARCHIVE processes all files of the specified volume.

(4) Since the NAME operand is not specified in the JOBVAR statement, ARCHIVE processes all job variables of the specified volume.

(5) The SAVE statement defines a save to MT cartridge.

- DIRECTORY=DIR.SAVES
  ARCHIVE is to write information on the save run to the directory file.
- VOLUME=QE0798
  This operand specifies the VSN of the save volume.
- DEVICE=TAPE-C4
  This operand specifies the device type for the MT cartridge.
- DUPLICATE=NO
  All duplicate files and job variables are saved only once, i.e. only the first file/job variable specified in the FILES or JOBVAR statement is written to the output volume. `DUPLICATE=NO` can only be specified if a directory file is used.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT. By default the report is output to SYSLST.

(6) Report of the save run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

7.2.5 Example 14: Saving individual file generations

Saving file generations from different file generation groups with a directory file from a public volume to magnetic tape or MT cartridge.

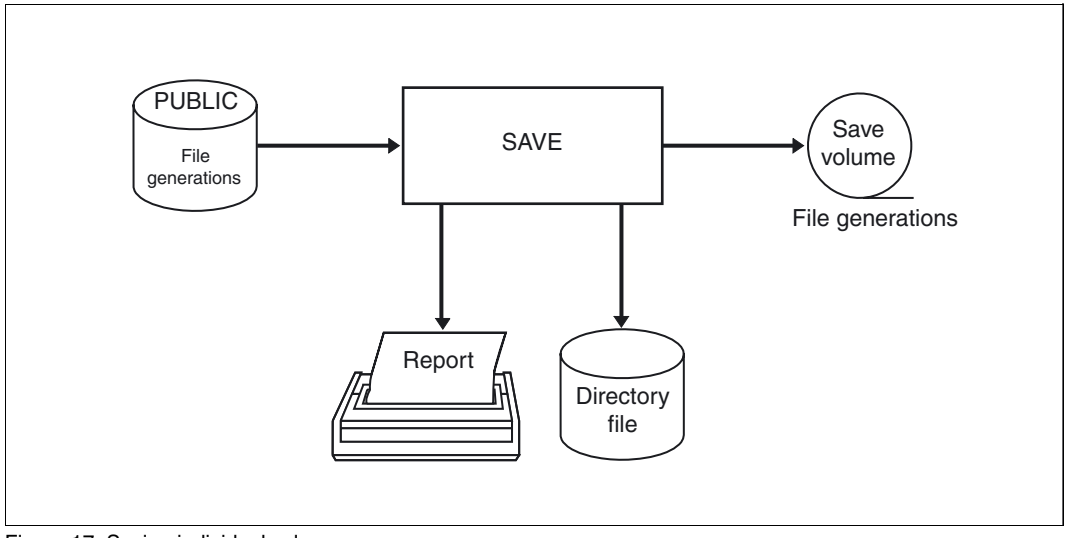


Figure 17: Saving individual volumes

```
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*FILES NAME=(FGG.1(*5),FGG.2(*63),FGG.3(*39)) ------------------------------  (1)
*SAVE DIRECTORY=DIR.FGG,NEW,VOLUME=QE0798,DEVICE=TAPE-C4,CHANGED=NO,
      LIST=SYSOUT ------------------------------------------------------------  (2)
*         S A V E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = NO
STREAM  = YES

FILES NAME=(FGG.1(*5),FGG.2(*63),FGG.3(*39))
SAVE DIRECTORY=DIR.FGG,NEW,VOLUME=QE0798,DEVICE=TAPE-C4,CHANGED=NO,
     LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100818.164733',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'ORKI' GENERATED
%  MARM127 VOLUME 'QE0798' RESERVED FOR USER ID 'TSOS'
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '91' PAM PAGES FOR '3' FILES
           AND '0' JVS IN '1' SECONDS
 ***         S A V E   A R C H I V E   R E P O R T ***         10-08-18       PAGE    1   (3)
                    SAVE VERSION IDENTIFIER - S.100818.164733 - ON VOLUME TYPE   TAPE-C4
SUBSAVE
 NUMBER       VSNS

    0        QE0798
      ***          S A V E   A R C H I V E   R E P O R T  ***         10-08-18         PAGE    2
                   SAVE VERSION IDENTIFIER - S.100818.164733 - ON VOLUME TYPE   TAPE-C4
                        ***   USER - TSOS       ***
FILE/JOB VARIABLE NAME                 LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                              VERS       SIZE  TYPE    VSN  TYP SAVE  VSN(S)

FGG.1(*0005)                     1         25  FULL  POST.2  D     0  QE0798
FGG.2(*0063)                     1         40  FULL  POST.2  D     0  QE0798
FGG.3(*0039)                     1         25  FULL  POST.1  D     0  QE0798
   ***     E N D   O F   A R C H I V E   R E P O R T  ***         10-08-18         PAGE    3

%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) The NAME operand in the FILES statement specifies the files to be processed in this ARCHIVE run.

(2) The SAVE statement defines a save to tape.

- DIRECTORY=DIR.FGG,NEW
  ARCHIVE is to write the information on the save run to the directory file. The `NEW` operand must be specified here, as the directory file is to be created as a new file. `NEW` must also be specified if the directory file has already been cataloged but does not yet have any ARCHIVE entries.
- TAPES=QE0798
  The corresponding MT cartridge is specified as the save tape and added to the volume pool of the directory file.
- DEVICE=TAPE-C4
  This operand specifies the device type for the MT cartridge.
- CHANGED=NO
  The files specified are saved in their entirety.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT. By default the report is output to SYSLST.

(3) Report of the save run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

7.3 Examples of system saves

7.3.1 Example 15: Full save

Performing a full save of the pubset with a directory file.

Step 1: Creating a directory file and setting up the volume pool

```
/SET-LOGON-PARAMETERS TSOS ──────────────────────────────────────────  (1)

/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*POOL DIRECTORY=DIR.81K,NEW,ADD=(QE0820,QE0821,QE0822,QE0823,QE0824,
                                 QE0825),DEVICE=TAPE-C4 ──────────────  (2)
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100820.100750',
           VERSION '9.0A'
%  MARM121 MAREN CATALOG ENTRY 'QE0820'/'0001' MODIFIED
%  ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'QE0820' ADDED TO THE POOL
%  MARM121 MAREN CATALOG ENTRY 'QE0820'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0821'/'0001' MODIFIED
%  ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'QE0821' ADDED TO THE POOL
%  MARM121 MAREN CATALOG ENTRY 'QE0821'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0822'/'0001' MODIFIED
%  ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'QE0822' ADDED TO THE POOL
%  MARM121 MAREN CATALOG ENTRY 'QE0822'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0823'/'0001' MODIFIED
%  ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'QE0823' ADDED TO THE POOL
%  MARM121 MAREN CATALOG ENTRY 'QE0823'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0824'/'0001' MODIFIED
%  ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'QE0824' ADDED TO THE POOL
%  MARM121 MAREN CATALOG ENTRY 'QE0824'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0825'/'0001' MODIFIED
%  ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'QE0825' ADDED TO THE POOL
%  ARC0003 ARCHIVE STATEMENT COMPLETED
```

(1) Open a dialog under the systems support ID (TSOS privilege).

(2) The directory file `DIR.81K` is created with the POOL statement.
The specified MT cartridges are assigned to your volume pool.

Step 2: Save run

```
*PARAM CATID=YES ————————————————————————————————————————————— (3)
%  ARC0017 PARAM STATEMENT ACCEPTED
*FILES NAME=:81K: ———————————————————————————————————————————— (4)
*SAVE DIRECTORY=DIR.81K,CHANGED=NO,TAPES=POOL,DEVICE=TAPE-C4,DRIVES=3
      LIST=SYSOUT,SHARE=YES ——————————————————————————————————— (5)
*         S A V E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = YES
STREAM  = YES

FILES NAME=:81K:
SAVE DIRECTORY=DIR.81K,CHANGED=NO,TAPES=POOL,DEVICE=TAPE-C4,DRIVES=3,
     LIST=SYSOUT,SHARE=YES
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100820.100758',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'OATT' GENERATED ————————————————— (6)
%  ARC0033 ARCHIVE SUBTASK TSN 'OATU' GENERATED
%  ARC0033 ARCHIVE SUBTASK TSN 'OATV' GENERATED
%  MARM121 MAREN CATALOG ENTRY 'QE0820'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0821'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0822'/'0001' MODIFIED
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '76364' PAM PAGES FOR '67' FILES
           AND '0' JVS IN '76' SECONDS
%  ARC0815 SUBTASK '1' HAS TRANSFERRED '76362' PAM PAGES FOR '65' FILES
           AND '0' JVS IN '75' SECONDS
%  ARC0815 SUBTASK '2' HAS TRANSFERRED '98087' PAM PAGES FOR '236' FILES
           AND '0' JVS IN '92' SECONDS
 ***         S A V E   A R C H I V E   R E P O R T  ***          10-08-20        PAGE    1  (7)
                    SAVE VERSION IDENTIFIER - S.100820.100758 - ON VOLUME TYPE   TAPE-C4
SUBSAVE
 NUMBER       VSNS

    0        QE0821
    1        QE0822
    2        QE0820
   ***          S A V E   A R C H I V E   R E P O R T  ***          10-08-20         PAGE    2
                    SAVE VERSION IDENTIFIER - S.100820.100758 - ON VOLUME TYPE   TAPE-C4
                    ***   CATALOG - 81K       USER - PRU        ***  ——————————————————— (8)
FILE/JOB VARIABLE NAME                  LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                              VERS        SIZE  TYPE    VSN  TYP SAVE  VSN(S)

ARCHIVE.P.100820.093827          1           3  FULL  81K.01  D     2  QE0820
FGG1                             0           0  FGGI          C     2  QE0820
FGG1(*0004)                      1        1000  FULL  81K.00  D     2  QE0820
```

```
FGG1(*0005)                              1        845  FULL  81K.00  D     2  QE0820
FGG1(*0006)                              1        840  FULL  81K.00  D     2  QE0820
FGG1(*0007)                              1        774  FULL  81K.00  D     2  QE0820
FGG1(*0008)                              1        774  FULL  81K.00  D     2  QE0820
PRU.BSP.E17                              1          1  FULL  81K.00  D     2  QE0820
PRU.BSP.17                               1          2  FULL  81K.00  D     2  QE0820
PRU.C1                                   1         86  FULL  81K.02  D     2  QE0820
PRU.C2                                   1         86  FULL  81K.02  D     2  QE0820
PRU.E                                    1          1  FULL  81K.00  D     2  QE0820
PRU.LST                                  1          2  FULL  81K.00  D     2  QE0820
PRU.0723.03                              1          2  FULL  81K.00  D     2  QE0820
PRU.0723.04                              1          3  FULL  81K.01  D     2  QE0820
PRU.0803.01                              1          3  FULL  81K.00  D     2  QE0820
TEST.01                                  1          4  FULL  81K.00  D     2  QE0820
TEST.02                                  1         20  FULL  81K.00  D     2  QE0820
TEST.03                                  1         18  FULL  81K.00  D     2  QE0820
TEST.04                                  1          4  FULL  81K.01  D     2  QE0820
TESTL                                    1          9  FULL  81K.01  D     2  QE0820
XX.FGG1                                  0          0  FGGI          C     2  QE0820
XX.FGG1(*0004)                           1       1000  FULL  81K.02  D     2  QE0820
XX.FGG1(*0005)                           1          2  FULL  81K.01  D     2  QE0820
XX.FGG1(*0006)                           1          4  FULL  81K.02  D     2  QE0820
XX.FGG1(*0007)                           1          3  FULL  81K.02  D     2  QE0820
XX.FGG1(*0008)                           1          6  FULL  81K.02  D     2  QE0820
                ***          S A V E   A R C H I V E   R E P O R T  ***          10-08-20          PAGE    3
                                SAVE VERSION IDENTIFIER - S.100820.100758 - ON VOLUME TYPE   TAPE-C4
                        ***   CATALOG - 81K       USER - PRUA       ***
           FILE/JOB VARIABLE NAME                    LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                                           VERS        SIZE  TYPE    VSN  TYP SAVE  VSN(S)

FALAST.2.FILE.001.OBS1                     1        1000  FULL  81K.00  D     0  QE0821
FALAST.2.FILE.001.OBS2                     1        2000  FULL  81K.00  D     0  QE0821
FALAST.2.FILE.001.OBS8                     1        1000  FULL  81K.02  D     0  QE0821
FALAST.2.FILE.001.OBTA                     1        1000  FULL  81K.02  D     0  QE0821
FALAST.2.FILE.001.OBTC                     1         845  FULL  81K.02  D     0  QE0821
FALAST.2.FILE.001.OBTE                     1         840  FULL  81K.01  D     0  QE0821
FALAST.2.FILE.001.OBTF                     1         774  FULL  81K.02  D     0  QE0821
FALAST.2.FILE.001.OBTG                     1         774  FULL  81K.00  D     0  QE0821
FALAST.2.FILE.001.OBTH                     1         625  FULL  81K.00  D     0  QE0821
FALAST.2.FILE.001.OBTI                     1         626  FULL  81K.01  D     0  QE0821
FALAST.2.FILE.001.OBTJ                     1         625  FULL  81K.02  D     0  QE0821
FALAST.2.FILE.002.OBS9                     1        2000  FULL  81K.00  D     0  QE0821
FALAST.2.FILE.002.OBTB                     1        2000  FULL  81K.00  D     0  QE0821
FALAST.2.FILE.002.OBTD                     1         775  FULL  81K.01  D     0  QE0821
FALAST.2.FILE.002.OBTK                     1         626  FULL  81K.02  D     0  QE0821
FA33.LIB                                   1        3376  FULL  81K.00  D     0  QE0821
                ***          S A V E   A R C H I V E   R E P O R T  ***          10-08-20          PAGE    4
                                SAVE VERSION IDENTIFIER - S.100820.100758 - ON VOLUME TYPE   TAPE-C4
                        ***   CATALOG - 81K       USER - PRUB       ***
           FILE/JOB VARIABLE NAME                    LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                                           VERS        SIZE  TYPE    VSN  TYP SAVE  VSN(S)

FALAST.2.FILE.001.OBS1                     1        1000  FULL  81K.01  D     1  QE0822
FALAST.2.FILE.001.OBS2                     1        2000  FULL  81K.01  D     1  QE0822
FALAST.2.FILE.001.OBS8                     1        1000  FULL  81K.01  D     1  QE0822
FALAST.2.FILE.001.OBTA                     1        1000  FULL  81K.01  D     1  QE0822
FALAST.2.FILE.001.OBTC                     1         845  FULL  81K.00  D     1  QE0822
FALAST.2.FILE.001.OBTE                     1         840  FULL  81K.01  D     1  QE0822
FALAST.2.FILE.001.OBTF                     1         774  FULL  81K.02  D     1  QE0822
FALAST.2.FILE.001.OBTG                     1         774  FULL  81K.01  D     1  QE0822
FALAST.2.FILE.001.OBTH                     1         625  FULL  81K.02  D     1  QE0822
FALAST.2.FILE.001.OBTI                     1         626  FULL  81K.01  D     1  QE0822
FALAST.2.FILE.001.OBTJ                     1         625  FULL  81K.02  D     1  QE0822
FALAST.2.FILE.002.OBS9                     1        2000  FULL  81K.01  D     1  QE0822
FALAST.2.FILE.002.OBTB                     1        2000  FULL  81K.01  D     1  QE0822
FALAST.2.FILE.002.OBTD                     1         775  FULL  81K.02  D     1  QE0822
```

```
FALAST.2.FILE.002.OBTK                        1        626  FULL  81K.00  D     1  QE0822
FA33.LIB                                      1       3376  FULL  81K.00  D     1  QE0822
                ***          S A V E   A R C H I V E   R E P O R T  ***         10-08-20         PAGE    5
                                SAVE VERSION IDENTIFIER - S.100820.100758 - ON VOLUME TYPE   TAPE-C4
                      ***   CATALOG - 81K       USER - PRUC       ***
          FILE/JOB VARIABLE NAME                    LASTPG/  SAVE    INPUT DEV  SUB  OUTPUT
                                          VERS        SIZE  TYPE     VSN  TYP SAVE  VSN(S)

FALAST.2.FILE.001.OBS1                        1       1000  FULL  81K.00  D     2  QE0820
FALAST.2.FILE.001.OBS2                        1       2000  FULL  81K.00  D     2  QE0820
FALAST.2.FILE.001.OBS8                        1       1000  FULL  81K.01  D     2  QE0820
FALAST.2.FILE.001.OBTA                        1       1000  FULL  81K.02  D     2  QE0820
FALAST.2.FILE.001.OBTC                        1        845  FULL  81K.00  D     2  QE0820
FALAST.2.FILE.001.OBTE                        1        840  FULL  81K.02  D     2  QE0820
FALAST.2.FILE.001.OBTF                        1        774  FULL  81K.00  D     2  QE0820
FALAST.2.FILE.001.OBTG                        1        774  FULL  81K.02  D     2  QE0820


. . .
further output lines omitted
. . .

                ***          S A V E   A R C H I V E   R E P O R T  ***         10-08-20         PAGE   18
                                SAVE VERSION IDENTIFIER - S.100820.100758 - ON VOLUME TYPE   TAPE-C4
                      ***   CATALOG - 81K       USER - TSOS       ***
          FILE/JOB VARIABLE NAME                    LASTPG/  SAVE    INPUT DEV  SUB  OUTPUT
                                          VERS        SIZE  TYPE     VSN  TYP SAVE  VSN(S)

PRU.0703.01                                   1          3  FULL  81K.00  D     2  QE0820
PRU.0703.02                                   1          3  FULL  81K.01  D     2  QE0820
PRU.0703.03                                   1          1  FULL  81K.02  D     2  QE0820
PRU.0703.08                                   1          1  FULL  81K.00  D     2  QE0820
PRU.0703.10                                   1          4  FULL  81K.01  D     2  QE0820
PRU.0705.B                                    1         45  FULL  81K.02  D     2  QE0820
PRU.0705.B2                                   1         15  FULL  81K.00  D     2  QE0820
PRU.0705.G32                                  1          4  FULL  81K.01  D     2  QE0820
PRU.0705.NK4                                  1         15  FULL  81K.02  D     2  QE0820
PRU.0705.SMP                                  1         35  FULL  81K.00  D     2  QE0820
PRU.0727.01                                   1         21  FULL  81K.01  D     2  QE0820
PRU.0730.04                                   1         20  FULL  81K.02  D     2  QE0820
PRU.0802.01                                   1          9  FULL  81K.00  D     2  QE0820
PRU.0802.02                                   1          4  FULL  81K.01  D     2  QE0820
PRU.0802.04                                   1          2  FULL  81K.02  D     2  QE0820
PRU.0802.05                                   1          2  FULL  81K.00  D     2  QE0820
PRU.0802.06                                   1          5  FULL  81K.01  D     2  QE0820
PRU.0802.07                                   1          3  FULL  81K.02  D     2  QE0820
SYSCAT.GUARDS                                 2          6  FULL  81K.02  D     2  QE0820
SYSSRPM.BACKUP                                1         21  FULL  81K.02  D     2  QE0820
                ***      E N D   O F   A R C H I V E   R E P O R T  ***         10-08-20         PAGE   19
```

```
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(3) This statement specifies that the catalog ID is to be used.

(4) In the following ARCHIVE run, all files in the pubset with the catalog ID `81K` are to be processed. Since the NAME operand is not specified, all user IDs are saved.

(5) The SAVE statement defines a save to tape.

- DIRECTORY=DIR.81K
  ARCHIVE is to write the information on the save run to the directory file.
- CHANGED=NO
  All files are always saved. The operand `CHANGED=NO` must be specified for a full save.
- TAPES=POOL
  The VSNs of the save volumes required are read from the pool of the specified directory file.
- DEVICE=TAPE-C4
  `TAPE-C4` type MT cartridges are used for the save run. This specification is necessary despite the POOL entry.
- DRIVES=3
  To speed up the save run 3 parallel runs are defined via the DRIVES operand. All the files of a user ID are saved in one run as a “packet”. Three save runs are started in parallel. Once one run has terminated, all the files of the next user ID
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT. By default the report is output to SYSLST.
- SHARE=YES
  The volumes and save files created are “shareable”.

(6) ARCHIVE starts three subtasks for the input/output.

(7) Report of the save run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.
The number of parallel runs is listed on page 1 of the report together with the associated VSNs.

(8) All the files of the user `PRU` are written as a “packet” to the tape with the volume serial number `QE0820` in one run.
Parallel to this run the files of `PRUA` are written to `QE0821` and those of `PRUB` to `QE0822`.
The subtasks each save one user ID as a packet in parallel and request the next packet (the next user ID) to be saved once the last file has been saved (see section “Parallel and serial processing” on page 75).

7.3.2 Example 16: Incremental save

Performing an incremental save of the pubset with a directory file.

```
/SET-LOGON-PARAMETERS TSOS ——————————————————————————————————————  (1)

/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*PARAM CATID=YES ————————————————————————————————————————————————  (2)
%  ARC0017 PARAM STATEMENT ACCEPTED
*FILES NAME=:81K: ———————————————————————————————————————————————  (3)
*SAVE DIRECTORY=DIR.81K,CHANGED=YES,TAPES=POOL,DEVICE=TAPE-C4,DRIVES=2,
      LIST=SYSOUT,RETPD=4,SHARE=YES ————————————————————————————————  (4)
*         S A V E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = YES
STREAM  = YES

FILES NAME=:81K:
SAVE DIRECTORY=DIR.81K,CHANGED=YES,TAPES=POOL,DEVICE=TAPE-C4,DRIVES=2,
     LIST=SYSOUT,RETPD=4,SHARE=YES
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100820.101338',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'OATX' GENERATED
%  ARC0033 ARCHIVE SUBTASK TSN 'OATY' GENERATED
%  MARM121 MAREN CATALOG ENTRY 'QE0823'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0824'/'0001' MODIFIED
%  ARC0810 THE RETENTION PERIOD OF SOME BACKUPS WILL BE MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0822'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0820'/'0001' MODIFIED
%  ARC0807 SAVE FILE 'S.100820.100758' RETENTION PERIOD UPDATED
%  MARM121 MAREN CATALOG ENTRY 'QE0821'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0822'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0820'/'0001' MODIFIED
%  ARC0807 SAVE FILE 'S.100820.100758' RETENTION PERIOD UPDATED
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '109' PAM PAGES FOR '2' FILES
           AND '0' JVS IN '2' SECONDS
```

```
            %  ARC0815 SUBTASK '1' HAS TRANSFERRED '52' PAM PAGES FOR '4' FILES
                       AND '0' JVS IN '0' SECONDS
             ***         S A V E   A R C H I V E   R E P O R T  ***          10-08-20        PAGE    1    (5)
                                SAVE VERSION IDENTIFIER - S.100820.101338 - ON VOLUME TYPE   TAPE-C4
          SUBSAVE
           NUMBER       VSNS

              0        QE0823
              1        QE0824
                ***          S A V E   A R C H I V E   R E P O R T  ***          10-08-20          PAGE    2
                                SAVE VERSION IDENTIFIER - S.100820.101338 - ON VOLUME TYPE   TAPE-C4
                     ***   CATALOG - 81K       USER - PRU        ***
          FILE/JOB VARIABLE NAME                    LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                                              VERS      SIZE  TYPE    VSN  TYP SAVE  VSN(S)

ARCHIVE.P.100820.093827                          1         3        81K.01  D     0  IN CATALOG BUT NOT SAVED
PRU.BSP.E17                                      1         1        81K.00  D     0  IN CATALOG BUT NOT SAVED
PRU.BSP.17                                       1         2        81K.00  D     0  IN CATALOG BUT NOT SAVED
PRU.C1                                           1        86        81K.02  D     0  IN CATALOG BUT NOT SAVED
PRU.C2                                           1        86        81K.02  D     0  IN CATALOG BUT NOT SAVED
PRU.E                                            1         1        81K.00  D     0  IN CATALOG BUT NOT SAVED
PRU.LST                                          1         2        81K.00  D     0  IN CATALOG BUT NOT SAVED
PRU.0723.03                                      1         2        81K.00  D     0  IN CATALOG BUT NOT SAVED
PRU.0723.04                                      1         3        81K.01  D     0  IN CATALOG BUT NOT SAVED
PRU.0803.01                                      1         3        81K.00  D     0  IN CATALOG BUT NOT SAVED
TEST.01                                          1         4        81K.00  D     0  IN CATALOG BUT NOT SAVED
TEST.02                                          1        20        81K.00  D     0  IN CATALOG BUT NOT SAVED
TEST.03                                          1        22  FULL  81K.00  D     0  QE0823
TEST.04                                          1         4        81K.01  D     0  IN CATALOG BUT NOT SAVED
TEST.05                                          1        86  FULL  81K.00  D     0  QE0823
TESTL                                            1         9        81K.01  D     0  IN CATALOG BUT NOT SAVED
XX.FGG1                                          0         0                C     0  IN CATALOG BUT NOT SAVED
XX.FGG1(*0004)                                   1      1000        81K.02  D     0  IN CATALOG BUT NOT SAVED

. . .
further output lines omitted
. . .

            %  ARC0003 ARCHIVE STATEMENT COMPLETED
            *END
            %  ARC0009 ARCHIVE TERMINATED
```

(1) Open a dialog under the systems support ID (`TSOS` privilege).

(2) This statement specifies that the catalog ID is to be used.

(3) In the following ARCHIVE run, all files of the pubset `81K` are to be processed.

(4) The SAVE statement defines a save to tape.

- DIRECTORY=DIR.81K
  ARCHIVE is to write the information on the save run to the existing directory file.
- CHANGED=YES
  Only those files are saved that are either not listed in the directory file or are listed there but have been changed since the last full or incremental save.

  The CHANGED operand is not required, since `CHANGED=YES` is there default. The same is true for the `BLOCK-SIZE=MAX` operand.
- TAPES=POOL
  The VSNs of the output tapes are drawn in ascending order from the pool of the directory file `DIR.FULL`.
- DEVICE=TAPE-C4
  This operand specifies the device type for the MT cartridge.
- DRIVES=2
  2 parallel runs are defined for this save. As considerably fewer files are saved in an incremental save than in a full save, a smaller number of parallel runs are required.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT. By default the report is output to SYSLST.
- RETPD=4
  The retention period for the save version is set to 4 days. The files of this save version cannot be deleted during this period.
- SHARE=YES
  The volumes and save files created are “shareable”.

(5) Report of the save run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

7.3.3 Example 17: Restoration from system save

Writing back files from previous full and incremental saves by systems support under the user ID.

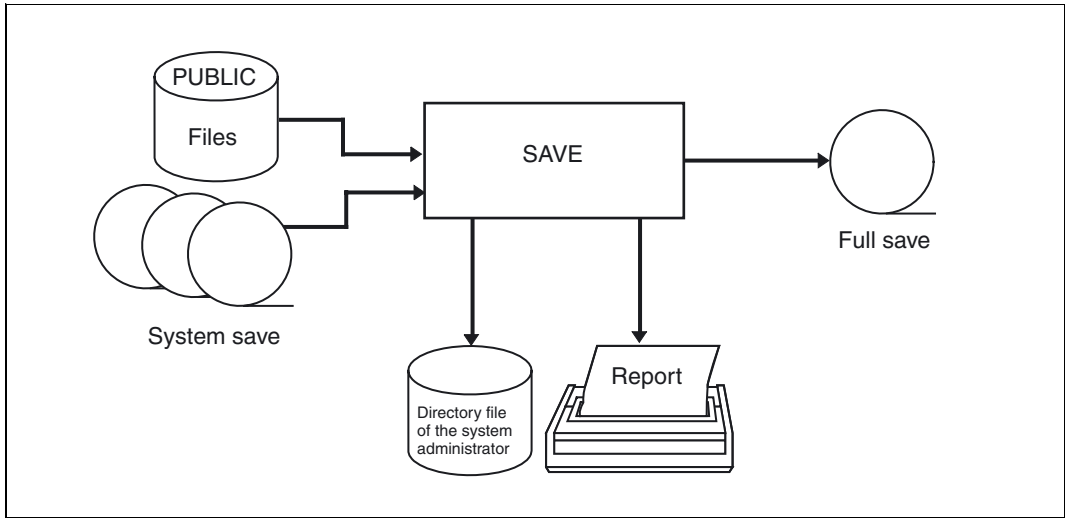


Figure 18: Restoration from system save

```
/SET-LOGON-PARAMETERS PRU ──────────────────────────────────────────────  (1)
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*PARAM CATID=YES ──────────────────────────────────────────────────────  (2)
%  ARC0017 PARAM STATEMENT ACCEPTED
*FILES NAME=TEST. ─────────────────────────────────────────────────────  (3)
*RESTORE DIRECTORY=$TSOS.DIR.81K,REPLACE=YES,LIST=SYSOUT ──────────────  (4)
*   R E S T O R E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
SNR     = YES
CATID   = YES
STREAM  = YES

FILES NAME=TEST.
RESTORE DIRECTORY=$TSOS.DIR.81K,REPLACE=YES,LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100820.101703',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'OAT1' GENERATED
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '138' PAM PAGES FOR '5' FILES
           AND '0' JVS IN '4' SECONDS
 ***   R E S T O R E   A R C H I V E   R E P O R T ***        10-08-20       PAGE    1  (5)
                      SAVE VERSION IDENTIFIER - S.100820.100758
SUBSAVE
 NUMBER       VSNS

    0        QE0820
      ***   R E S T O R E   A R C H I V E   R E P O R T ***        10-08-20        PAGE    2
                      SAVE VERSION IDENTIFIER - S.100820.101338
SUBSAVE
 NUMBER       VSNS

    0        QE0823
      ***   R E S T O R E   A R C H I V E   R E P O R T ***        10-08-20        PAGE    3
          ***   CATALOG - 81K       USER - PRU        ***
FILE/JOB VARIABLE NAME                LASTPG/   SAVE VERSION  SAVE  INPUT   SUB OUTPUT
                              VERS       SIZE    IDENTIFIER   TYPE   VSN   SAVE DISK(S)

TEST.01                          1          4  100820.100758  FULL  QE0820   0  81K.00
TEST.02                          1         20  100820.100758  FULL  QE0820   0  81K.00
TEST.03                          1         22  100820.101338  FULL  QE0823   0  81K.00
TEST.04                          1          4  100820.100758  FULL  QE0820   0  81K.00
TEST.05                          1         86  100820.101338  FULL  QE0823   0  81K.02
      ***     E N D   O F   A R C H I V E   R E P O R T ***        10-08-20        PAGE    4
```

```
%  ARC0003 ARCHIVE STATEMENT COMPLETED
*END
%  ARC0009 ARCHIVE TERMINATED
```

(1) Open a dialog under the nonprivileged user ID PRU.

(2) This statement specifies that the catalog ID is to be used.

(3) The NAME operand in the FILES statement specifies the files to be processed by ARCHIVE.

(4) The RESTORE statement instructs ARCHIVE to restore the files saved to tape.

- DIRECTORY=$TSOS.DIR.81K
  ARCHIVE is to restore the saved files from the directory file of the system support.
- REPLACE=YES
  If a file with the same name already exists under the PRU user ID, it is replaced on the save tape with this file.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT.
  By default the report is output to SYSLST.

(5) Report of the restoration run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.
The files TEST.01, TEST.02 and TEST.04 are taken from “Example 15: Full save” on page 242, the files TEST.03 and TEST.05 from “Example 16: Incremental save” on page 247.

7.3.4 Example 18: Creating a complete system save

Creating a complete current system save with a directory file from a previous full save (“Example 15: Full save” on page 242) and if necessary several incremental saves (“Example 16: Incremental save” on page 247).

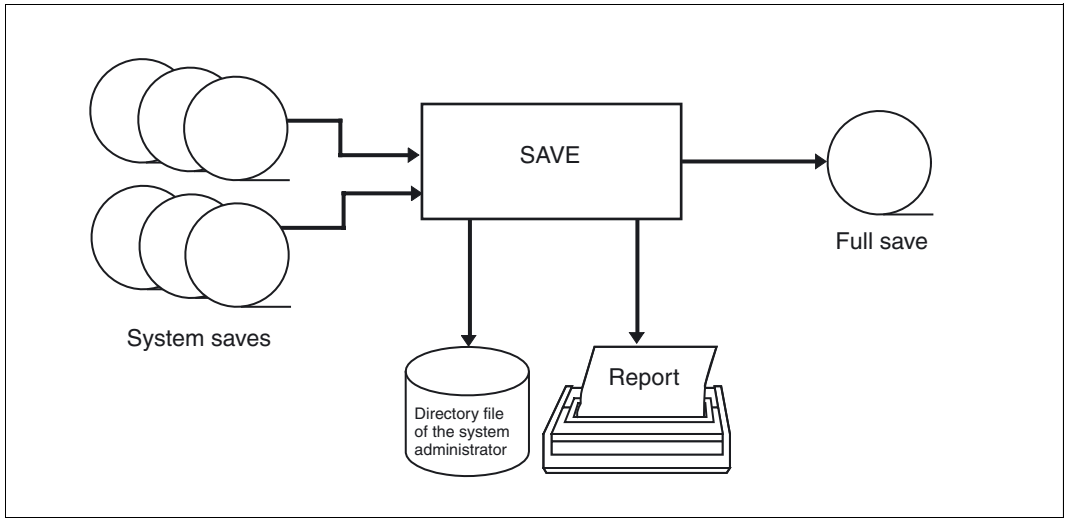


Figure 19: Creating a complete system save

```
/SET-LOGON-PARAMETERS TSOS ------------------------------------------------ (1)
/START-ARCHIVE
%  BLS0990 PROCESSING OF REP FILE ':SBZ3:$TSOS.SYSREP.ARCHIVE.090' STARTED
%  ARCLOAD Program 'ARCHIVE', version '09.0A' of '2009-05-20' loaded
           from file ':1OSH:$TSOS.SYSLNK.ARCHIVE.090'
%  ARCCOPY Copyright (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
            All rights reserved
%  ARC0001 ARCHIVE READY
*PARAM CATID=YES ---------------------------------------------------------- (2)
%  ARC0017 PARAM STATEMENT ACCEPTED
*FILES FROM=S.100820.101338 ----------------------------------------------- (3)
*FILES FROM=S.100820.100758
*SAVE DIRECTORY=DIR.81K,TAPES=POOL,DEVICE=TAPE-C4,DUPLICATE=NO,
     LIST=SYSOUT ---------------------------------------------------------- (4)
*         S A V E   C O M M A N D   L I S T I N G ***

PARAMETER VALUES:-
CNS     = YES
UNLOAD  = NO
RESTART = YES
OPERATOR= NO
OLS     = NO
CATID   = YES
STREAM  = YES

FILES FROM=S.100820.101338
FILES FROM=S.100820.100758
SAVE DIRECTORY=DIR.81K,TAPES=POOL,DEVICE=TAPE-C4,DUPLICATE=NO,
     LIST=SYSOUT
%  ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.100820.160026',
           VERSION '9.0A'
%  ARC0033 ARCHIVE SUBTASK TSN 'OAUG' GENERATED
%  MARM121 MAREN CATALOG ENTRY 'QE0825'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0826'/'0001' MODIFIED
%  MARM121 MAREN CATALOG ENTRY 'QE0827'/'0001' MODIFIED
%  ARC0815 SUBTASK '0' HAS TRANSFERRED '250795' PAM PAGES FOR '372' FILES
           AND '0' JVS IN '77' SECONDS
 ***         S A V E   A R C H I V E   R E P O R T ***         10-08-20       PAGE    1   (5)
                    SAVE VERSION IDENTIFIER - S.100820.160026 - ON VOLUME TYPE   TAPE-C4
SUBSAVE
 NUMBER       VSNS

   0        QE0825  QE0826  QE0827
     ***          S A V E   A R C H I V E   R E P O R T ***         10-08-20         PAGE    2
                  SAVE VERSION IDENTIFIER - S.100820.160026 - ON VOLUME TYPE   TAPE-C4
          ***   CATALOG - 81K       USER - PRU        ***
```

```
          FILE/JOB VARIABLE NAME                  LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                                        VERS        SIZE  TYPE    VSN  TYP SAVE  VSN(S)

ARCHIVE.P.100820.093827                    1           3  FULL  QE0820  T     0  QE0826
FGG1                                       0           0  FGGI  QE0823  T     0  QE0825
FGG1(*0004)                                1        1000  FULL  QE0823  T     0  QE0825
FGG1(*0005)                                1         845  FULL  QE0823  T     0  QE0825
FGG1(*0006)                                1         840  FULL  QE0823  T     0  QE0825
FGG1(*0007)                                1         774  FULL  QE0823  T     0  QE0825
FGG1(*0008)                                1         774  FULL  QE0823  T     0  QE0825
PRU.BSP.E17                                1           1  FULL  QE0820  T     0  QE0826
PRU.BSP.17                                 1           2  FULL  QE0820  T     0  QE0826
PRU.C1                                     1          86  FULL  QE0820  T     0  QE0826
PRU.C2                                     1          86  FULL  QE0820  T     0  QE0826
PRU.E                                      1           1  FULL  QE0820  T     0  QE0826
PRU.LST                                    1           2  FULL  QE0820  T     0  QE0826
PRU.0723.03                                1           2  FULL  QE0820  T     0  QE0826
PRU.0723.04                                1           3  FULL  QE0820  T     0  QE0826
PRU.0803.01                                1           3  FULL  QE0820  T     0  QE0826
TEST.01                                    1           4  FULL  QE0820  T     0  QE0826
TEST.02                                    1          20  FULL  QE0820  T     0  QE0826
TEST.03                                    1          22  FULL  QE0823  T     0  QE0825
TEST.04                                    1           4  FULL  QE0820  T     0  QE0826
TEST.05                                    1          86  FULL  QE0823  T     0  QE0825
TESTL                                      1           9  FULL  QE0820  T     0  QE0826
XX.FGG1                                    0           0  FGGI  QE0820  T     0  QE0826
XX.FGG1(*0004)                             1        1000  FULL  QE0820  T     0  QE0826
XX.FGG1(*0005)                             1           2  FULL  QE0820  T     0  QE0826
XX.FGG1(*0006)                             1           4  FULL  QE0820  T     0  QE0826
XX.FGG1(*0007)                             1           3  FULL  QE0820  T     0  QE0826
XX.FGG1(*0008)                             1           6  FULL  QE0820  T     0  QE0826
                ***          S A V E   A R C H I V E   R E P O R T  ***          10-08-20          PAGE    3
                                SAVE VERSION IDENTIFIER - S.100820.160026 - ON VOLUME TYPE   TAPE-C4
                     ***   CATALOG - 81K       USER - PRUA       ***
          FILE/JOB VARIABLE NAME                  LASTPG/  SAVE   INPUT DEV  SUB  OUTPUT
                                        VERS        SIZE  TYPE    VSN  TYP SAVE  VSN(S)

FALAST.2.FILE.001.OBS1                     1        1000  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.001.OBS2                     1        2000  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.001.OBS8                     1        1000  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.001.OBTA                     1        1000  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.001.OBTC                     1         845  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.001.OBTE                     1         840  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.001.OBTF                     1         774  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.001.OBTG                     1         774  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.001.OBTH                     1         625  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.001.OBTI                     1         626  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.001.OBTJ                     1         625  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.002.OBS9                     1        2000  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.002.OBTB                     1        2000  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.002.OBTD                     1         775  FULL  QE0821  T     0  QE0825
FALAST.2.FILE.002.OBTK                     1         626  FULL  QE0821  T     0  QE0825
FA33.LIB                                   1        3376  FULL  QE0821  T     0  QE0825

. . .

further output lines omitted

. . .

            %  ARC0003 ARCHIVE STATEMENT COMPLETED
            *END
            %  ARC0009 ARCHIVE TERMINATED
```

(1) Open a dialog under the systems support ID (TSOS privilege).

(2) This statement specifies that the catalog ID is to be used.

(3) FILES=S.100820.101338
FILES=S.100820.100758
These FILES statements specify that all files from the last full and incremental saves are to be processed. The save versions must be specified in descending order from the newest to the oldest.

(4) The SAVE statement defines a save to tape.

- DIRECTORY=DIR.81K
  ARCHIVE is to write the information about the save run to the existing directory file.
- TAPES=POOL
  The VSN of the output volumes are taken from the volume pool of the directory file `DIR.FULL` in ascending order.
- DEVICE=TAPE-C4
  This operand specifies the device type for the MT cartridge.
- DUPLICATE=NO
  All duplicated files are only saved once, i.e. only the first file entered in the FILES statement is written to the output volume. The `DUPLICATE=NO` operand is only permitted for the directory file.
- LIST=SYSOUT
  ARCHIVE is to output the report of the run to SYSOUT.
  By default the report is output to SYSLST.

(5) Report of the save run. The processed files are listed. The meanings of the columns are explained in section “ARCHIVE report” on page 52.

8 ARCHIVE macro

You can call ARCHIVE from user programs by means of the ARCHIVE macro. The statements in the macro are specified in the same way as in interactive mode (free string format). However, the HELP, PARAM and END statements are not permitted. In the macro, the parameters can be specified with their own operands.

```
ARCHIVE

address,length[,MF = L / (E,opadr) / (E,(1)) ]
,CNS = YES  /  NO
,RES = YES  /  NO
,UNL = NO  /  YES
,OP = {NO  /  YES
,WRCHK = NO  /  YES
,SNR = YES  /  NO
,DEST = NO  /  YES
,CATID = NO  /  YES
,STREAM = YES /  NO
,OLS = NO  /  YES
```

address
Symbolic address of the ARCHIVE statement.

length
Length of the ARCHIVE statement.

MF
An operand list is generated.
For a description see the manual “Executive Macros” [5].

The operands of the ARCHIVE macro are similar to those of the PARAM statement; there are only a few operands which do not have the same names as in the PARAM statement, namely: RES stands for RESTART, UNL for UNLOAD, OP for OPERATOR and DEST for DESTROY.

Return codes and error flags

Register 1 is overwritten.

The rightmost byte of register 15 contains the return code (RC), and the remaining bytes are deleted.

RC	Meaning
X'00'	The statement was executed without errors.
X'10'	The statement was executed with warnings.
X'20'	The statement was executed with errors.
X'40'	Execution of the statement was not completed after an error.
X'80'	The statement was rejected due to an error.
X'81'	The address list was entered incorrectly or the statement is invalid; execution was not started.
X'FF'	ARCHIVE is not loaded.

Job switches are not set when ARCHIVE is called via the ARCHIVE macro.

Example

```
ASSEMBH LISTING                                                                                    15:51:13  2010-08-13   PAGE 0002
          SYMBOL                          TYPE ID  ADDR     LENGTH  A/R-MODE                EXTERNAL SYMBOL DICTIONARY
          ARCHIVE                          SD 0001 00000000 0000A6  24  24

ASSEMBH LISTING                                                                                    15:51:13  2010-08-13   PAGE 0003
 LOCTN  OBJECT CODE    ADDR1    ADDR2     STMNT  M  SOURCE STATEMENT
 000000                                       1     ARCHIVE  START
 000000 05 40                                 2     BEGIN    BALR  4,0
 000002                 00000002              3              USING *,4
                                              4              ARCHIVE COMMAND,L'COMMAND ------------------------------------------ (1)
                                              5   1            IDMF  MF=
 000002 0700                                  6   1            CNOP  0,4                                                  |      |
                                              7   1            ##BAL   1,ID0001
 000004 4D 10 4036      00000038              8   2          BAS   1,ID0001                                          012
 000008 00000056                              9   1            DC    A(COMMAND)          STATEMENT ADDRESS                |      |
 00000C 0051                                 10   1            DC    AL2(80+1)           STATEMENT LENGTH                 |      |
 00000E 0000                                 11   1            DC    H'0'                VERSION OF MACRO                 |      |
 000010 0000                                 12   1           DC    B'0000000000000000'                                   |      |
 000012 0003                                 13   1           DC    B'0000000000000011'                                   |      |
 000014 0000000000000000                     14   1 SVID      DC    XL15'00'            RETURN SVID                      | 422  |
 000023 00                                   15   1           DC    2OX'00'                                              |      |
 000038                                      16   1 ID0001     DS    0H                                                  |      |
 000038 0A 83                                17   1           SVC   131                                                  |      |
                                             18              PRINT NOGEN
 00003A                                      19              TERM
 000056 C6C9D3C5E240D5C1                     38     COMMAND  DC    CL80'FILES NAME=FILE. SAVE DIRECTORY=ARCHIVE.DIR,VOLUME=* ---- (2)
 000056                                      38                    OP,DEVICE=TAPE-C4,L=SYSOUT'
                                             39              END
FLAGS IN 00000 STATEMENTS, 000 PRIVILEGED FLAGS, 000 MNOTES
HIGHEST ERROR-WEIGHT : NO ERRORS
THIS PROGRAM WAS ASSEMBLED BY ASSEMBH     V01.2D01     ON 2001-09-13 AT 15:51:11

ASSEMBH LISTING                                                                                    15:51:13  2010-08-13   PAGE 0004
USED FILES AND LIBRARIES
SOURCE LIBRARY  :      :2OSG:$TEST.MACEXMP.LIB
SOURCE ELEMENT  :      ARCHIVE
VERS/DATE       :      @/2010-08-13
MODULE LIBRARY  :      :2OSG:$TEST.MACEXMP.LIB ------------------------------------------------------------------------------------- (3)
MODULE ELEMENT  :      ARCHIVE
VERS/DATE       :      @/2010-08-13
MACRO-LIBRARIES     LINKNAME  LIBRARY-NAME
                              :2OSH:$TSOS.SYSLIB.ARCHIVE.090
                              :2OSH:$TSOS.MACROLIB
ASSEMBLY TIME :          0.243     SEC.
THIS LISTING WAS GENERATED BY THE LISTING GENERATOR V 1.2D01.
```

(1) In the ARCHIVE macro, the operands `COMMAND` and `L'COMMAND` specify the address and length respectively of the ARCHIVE statement in this Assembler program.

(2) One FILES statement and one SAVE statement are defined at the address `COMMAND`. The NAME operand in the FILES statement specifies the files with the partially qualified filename `FILE.`.
The SAVE statement specifies that ARCHIVE is to work with the directory file `ARCHIVE.DIR`. The save volume is taken from the volume pool, the volume type is specified.

(3) The program that was, for example, stored in a library as the element `ARCHIVE`, can now be activated with

```
/START-EXECUTABLE-PROGRAM
  FROM-FILE=*LIBRARY-ELEMENT(LIBRARY=$TEST.MACEXMP.LIB,
                             ELEMENT-OR-SYMBOL=ARCHIVE)
```

or, in abbreviated form:

```
/SRX LIB=$TEST.MACEXMP.LIB,ELEM=ARCHIVE
```

Glossary

alias name

ARCHIVE also permits alias names instead of file names. The assignments of file names to alias names are defined in the alias catalog. Internally, ARCHIVE deals with file names only.

ARCHIVE.CHKPT

→ checkpoint file

archive directory

→ directory file

ARCHIVE.LATER

→ queue file

ARCHIVE macro

ARCHIVE can be called from user programs by means of the ARCHIVE macro. In the macro, statements are specified in the same way as in interactive mode. However, the HELP, PARAM and END statements are not permitted.

ARCHIVE sequence number

An ARCHIVE process is created for statements requiring inputs/outputs to a file or to a save volume. This process is assigned a 15-digit ARCHIVE sequence number.

archiving

Swapping out of data no longer required online from disk to magnetic tape or MT cartridges.

backup

→ data saving

backup class

File save level entered in the catalog. Determines how often a file is saved. Possible values: A, B, C, D and E. Files with backup level A are always saved.

cataloged not saved
The file was not saved, either because it had not been updated when an → incremental save occurred or because it could not be saved due to an error (e.g. OPEN error).

CFID (Coded File ID)
→ internal file name

checkpoint file
Contains an entry for each current save version and, if the ARCHIVE run was effected with the RESTART=YES operand, the checkpoints at which an abnormally terminated ARCHIVE run can be restarted.

CNS
→ cataloged not saved

Coded File ID (CFID)
→ internal file name

command listing
For every ARCHIVE run, ARCHIVE outputs the entered statements in the form of a command listing.

conversion
ARCHIVE offers a conversion option in cases where files have a PAM key but the output volume does not permit the writing of PAM keys.

data
Within ARCHIVE: files, job variables and catalog entries of files stored on magnetic tape, MT cartridge or disk.

data saving
Within ARCHIVE, the periodic creation of copies of the entire data inventory for the purpose of → reconstruction of → data in case data is lost due to hardware errors, inadvertent deletion etc. Data saving can also be used to reorganize disk storages.

DIRCONV
The DIRCONV conversion program enables directory files to be converted from `CATID=NO` mode to `CATID=YES` mode. Conversion in the opposite direction is not possible.

directory file

Automatically maintained administrative directory of the saved data (files, job variables, → save files and → save versions) and of the related → volumes.

ENTER file

ARCHIVE creates this for a batch process in the → queue file after the batch process has been started via the PROCESS statement.

export

Writing → data to tape for → transfer.

full save

All files or job variables are saved completely, regardless of whether or not they have been updated since the last save run.

HSMS (Hierarchical Storage Management System)

BS2000/OSD software product with migration, → backup and archival functions; implemented via a storage hierarchyconcept and archives. HSMS is based on ARCHIVE functions.

import

Reading exported → data into the target server for → transfer.

incremental save

An incremental save includes only those files that have been updated or newly created since the last save run. This save method decreases runtime and reduces storage space requirements.

internal file name

Internal name in addition to the file name in the catalog; serves for unique identification of the file and changes whenever the file is updated. Is passed in the FSTAT macro but not output via `/SHOW-FILE-ATTRIBUTES`.

job switch

In procedures and ENTER jobs, information on the execution of an ARCHIVE process can be derived from the position of job switches set by ARCHIVE during or after a run.

LATER job

Within ARCHIVE, a job which is running in batch mode due to specification of the `NOW=NO` operand and which is written to the → queue file `ARCHIVE.LATER`. The LATER job can be started or deleted at a later point in time.

MAREN

Software product managing magnetic tapes and MT cartridges in data centers. The ARCHIVE → volume pools can be included in MAREN management. In this case MAREN assigns the volumes to an ARCHIVE directory file. The MAREN entry of the volume indicates the assignment to the → directory file.

parallel processing

Files or job variables specified in a single FILES or JOBVAR statement can be simultaneously written to or read from a number of save volumes.

partial save

The user can specify, in the case of large files, that only those pages should be saved that have been updated since the last → full save.

Performant File Access

Can be defined for directory, checkpoint and print files, i.e. for files continually accessed. This requires specification of the PERFORMANCE attributes for these files.

PFA

→ Performant File Access

pool

→ volume pool

print file

A print file is created for every ARCHIVE run with a → restart option if a value other than NONE has been specified for the LIST operand. It contains information on the run and an alphabetical list of the files processed in the current run.

queue file

ARCHIVE statements can be processed in a batch process if the operand `NOW=NO` is specified for the relevant statement. The batch process is entered in the queue file `ARCHIVE.LATER` and can be started or deleted at a later point in time.

read-after-write check

Such a check can be requested for writing to disk. It is not effected by ARCHIVE but by DMS. After every write operation, the written data is read so that any errors can be pinpointed immediately.

reconstruction

Restoration of data in cases of data loss due to a hardware error or inadvertent deletion. For this purpose, save copies are accessed which were created at an earlier point in time.

report

ARCHIVE issues a report for every ARCHIVE run. The report contains information on the input/output volumes, the processed files and job variables, the save type, the save version number, the user ID and the date of the ARCHIVE run.

restart

ARCHIVE optionally writes checkpoints to the → checkpoint file. These checkpoints permit ARCHIVE runs interrupted due to a system crash or other errors to be resumed as of a defined status.

save file

Container holding saved files and job variables; comprises one or more → save versions and consists of a set of → volumes with the same owner and retention period. A save file can only be released as a whole. It is identified by a → svid formed from the date and time of day.

save version

Files or job variables saved at a certain point in time are combined into a save version. A save version is identified internally by a → svid and can be addressed by the user via its creation date or via the name assigned to it during creation.

saving

General term for writing (copying) → data to a → save file, regardless of the function involved; also used specifically for → data saving.

saving, logical

Reading → data from one or more → volumes and writing it to one or more volumes as a logical entity.

saving, physical

Writing all the → data of a → volume, including the volume labels, block by block to another volume whose contents and structure are identical to the original volume.

serial processing

The various FILES statements are processed one after the other: all the files specified in one FILES statement are transferred before the next FILES statement is processed. The approach for JOBVAR statements is analogous.

subtask

Statements requiring input/output from or to save volumes are normally processed by a subtask created by ARCHIVE, and not by the user job (main task).

svid

Save version identifier designating a → save file. It has the format:
`S.yymmdd.hhmmss`

transfer

Transfer of files, job variables or catalog entries of files to other BS2000/OSD systems or other user IDs; implemented via → export to tape and → import to the target server by means of ARCHIVE.

volumes

Within ARCHIVE: magnetic tapes, MT cartridges and disks.

volume pool

Set of → volumes managed in the → directory file. The volumes for the save jobs are requested from the pool of free volumes unless otherwise specified.

work files

ARCHIVE work files can be divided into two categories:

– files under the TSOS user ID that exist only once in the system and are created under TSOS during the first run (→ checkpoint file and → queue file)
– files that are created under the caller's user ID during every ARCHIVE run (e.g. the → save files).

Abbreviations

A-U	Allocation Unit
ACL	Access Control List (SECOS up to V3.0)
ACS	Alias Catalog System
BACL	Basic Access Control List
bpi	bits per inch
CATID	Catalog ID
CFID	Coded File Identifier
CNS	Cataloged Not Saved
CPU	Central Processing Unit
DEFLUID	Default Userid
DIRCONV	DIRectory CONVersion routine
DMS	Data Management System
DSSM	Dynamic Subsystem Management
EOF	End of File Label
EOT	End of Tape
EOV	End of Volume Label
FARMTSAV	File Archiving Metadata Save
FBA	Fixed Block Architecture
FDDRL	Fast Disk Dump and Reload
FGG	File generation group
FITC	Fast Inter Task Communication
HDR	File Header Labe
HERS	Hardware Error Recovery System
HSMS	Hierarchical Storage Management System
IMON	Installation Monitor
INIT	INITialize tapes routine
INOP	Inoperable

I/O	Input/Output
ISAM	Indexed Sequential Access Method
JV	Jobvariable
LMS	Library Maintenance System
MAREN	Dienstprogramm zur Bandverarbeitung
MN	Mnemotechnical device name
MRSCAT	MRS Catalog
MT	Magnetic Tape
MTC	Magnetic Tape Cartridge
NDM	Nucleus Device Management
NK	None-Key disk format
NK2	None-Key 2K disk format
NK4	None-Key 4K disk format
PAM	Primary Access Method
PFA	Performant File Access
PLAM	Program Library Access Method
PTAM	Privileged Tape Access Method
PPAM	Privileged Primary Access Method
PPD	Protected Private Disk
Pubset	Public Volume Set
SAM	Sequential Access Method
SDF	System Dialog Facility
SFID	Save File Identification
SSCM	Static Subsystem Catalog Management
SVC	Supervisor Call
SVID	Save Version Identification
SVL	Standard Volume Label
TFT	Task File Table
TM	Tapemark
TPR	Task Privileged
TSN	Task Sequence Number
TSOS	Privilege or privileged user ID
TU	Task Unprivileged

UDS	Universal Database System
UHL	User File Header Label
UPAM	User Primary Access Method
UTL	User Trailer Label
VOL	Volume Header Label
VOLIN	VOLume INitialization routine
VSN	Volume Serial Number
VTOC	Volume Table of Contents

Related publications

The manuals listed below are available from these sources:

- On the Internet from our manual server (URL: *http://manuals.ts.fujitsu.com*)
- On the DVD “BS2000/OSD SoftBooks”.
  This CD contains practically all the manuals and readme files for the BS2000 system software of the latest BS2000/OSD version.

Printed copies of the manuals can be purchased separately over the Internet from: *http://manualshop.ts.fujitsu.com*.

[1] **BS2000/OSD-BC**
Utility Routines
User Guide

[2] **BS2000/OSD-BC**
Introductory Guide to DMS
User Guide

[3] **BS2000/OSD-BC**
Introductory Guide to Systems Support
User Guide

[4] **BS2000/OSD-BC**
Commands
User Guide

[5] **BS2000/OSD-BC**
Executive Macros
User Guide

[6] **B2000/OSD-BC**
System Installation
User Guide

[7] **DSSM/SSCM**
Subsystem Management in BS2000/OSD
User Guide

[8] **FDDRL** (BS2000/OSD)
User Guide

[9] **HSMS / HSMS-SV** (BS2000/OSD)
Hierarchical Storage Management System
Volume 1: Functions, Management and Installation
Volume 2: Statements
User Guide

[10] **IMON** (BS2000/OSD)
Installation Monitor
User Guide

[11] **JV** (BS2000/OSD)
Job Variables
User Guide

[12] **MAREN** (BS2000/OSD)
Volume 1: Basics of MTC Management
Volume 2: User Interfaces
User Guide

[13] **PERCON** (BS2000/OSD)
User Guide

[14] **SECOS** (BS2000/OSD
Security Control System
User Guide

Index

S