

BacPac User's Guide

Table of Contents

BacPac for OpenVMS	3
Introduction	3
Using this Help	3
Online Help	4
Installation	5
Installation Requirements	5
Installation Options	5
Selecting an Installation Directory	5
Conventions used during Installation	6
Installation Procedure	6
Installing BacPac as a known image	8
BacPac License Files	8
Updating a License file	9
BacPac Command Format	11
BacPac Command Syntax	11
Operations	11
Parameters	11
Qualifiers	12
BacPac Operations	13
Save Operations	13
Restore Operations	14
List Operations	15
Copy Operations	15
Duplicate Operations	16
Support	17

Getting Started	19
Verifying BacPac is Installed and Available.....	19
Creating a Pacset	19
Creating a Pacset on Tape	20
Creating a Pacset on a Remote Node	20
Listing a Pacset.....	20
Listing a Pacset on Multiple Tapes	21
Creating a BacPac Journal File	21
Listing a BacPac Journal File.....	22
Selecting files to be saved	22
Restoring a Pacset.....	23
Invoking BacPac from Within Batch Files	24
Translating an Exit Code.....	24
Using CTRL-T to Monitor Progress.....	25
Setting the Process Priority Using /PRIORITY	25
Image Mode BacPac.....	27
Image Mode Save.....	27
Image Mode Restore	27
Incremental and Differential Backups	29
Incremental Save	29
Differential Save.....	29
Incremental Restore.....	30
Differential Restore	30
Remote BacPac	31
Remote BacPac	31
Saving files to a remote node	31

Saving files from a remote node	31
Listing a remote Pacset	32
Restoring files to a local node.....	32
Restoring files to a remote node.....	32
Image, Incremental and Differential modes	33
BacPac Duplication	35
BacPac Duplication.....	35
Copying Files to/from a Remote Node.....	35
Duplicating a local disk to/from a remote node.....	36
Mounting the Target Disk.....	37
Duplication and Incremental Backups	37
Using Compression.....	38
Encryption	39
Encryption Keys and Algorithms.....	39
Using Encryption.....	39
Tape Handling.....	41
Tape Labels	41
Default Tape Labels.....	41
Tape Security.....	42
OPCOM Messages	42
Multi Volume Tape Pacsets	43
Copying Tape Pacsets.....	43
Disaster Recovery License	45
Disaster Recovery License	45
Qualifiers	47
Qualifiers Reference List.....	47

/ALIAS	47
/ASSIST	48
/BACKUP	49
/BEFORE	51
/BLOCK_SIZE	52
/BRIEF	53
/BROADCAST	54
/BY_OWNER	55
/CBT	56
/COMMENT	57
/COMPRESSION	58
/CONFIRM	59
/CONTIGUOUS	60
/COPY	61
/CREATED	62
/DELETE	63
/DENSITY	64
/DIFFERENTIAL	65
/DISMOUNT	66
/DUPLICATE	67
/ENCRYPT	68
/EXCEPTIONS	70
/EXCLUDE	71
/EXPIRED	72
/FULL	73
/IGNORE	74

/IMAGE.....	76
/INCREMENTAL.....	77
/INITIALIZE.....	78
/JOURNAL.....	79
/KEEP.....	80
/LABEL.....	81
/LIST.....	83
/LOCK_DEVICE.....	84
/LOG.....	85
/MEDIA_FORMAT.....	85
/MODIFIED.....	86
/MOUNT_FOREIGN.....	87
/NAME.....	88
/NEW_VERSION.....	89
/NONE.....	91
/NOTIFY.....	92
/OVERLAY.....	93
/PACSET.....	94
/PASSWORD.....	94
/PRIORITY.....	95
/PROTECTION.....	96
/RECORD.....	97
/RELEASE_TAPE.....	98
/REMOUNT.....	99
/REPLACE.....	100
/REWIND.....	101

/SELECT	102
/SINCE	103
/STATISTICS	104
/SUMMARY	105
/TAPE_EXPIRATION	106
/TRUNCATE	107
/VERIFY	108
/VERSION	109
/VOLUME_NAME	110
/ZERO_DATA	111
Index	113

The information contained in this manual is subject to change without notice. Compact Data Works, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular use. Compact Data Works, Inc. shall not be liable for errors contained herein nor for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied or reproduced without the prior written consent of Compact Data Works, Inc.

Trademarks

BacPac is a trademark of Compact Data Works, Inc.

Alpha AXP, DEC, DECnet, VAX, VAXcluster, VMS, OpenVMS are registered trademarks of Hewlett-Packard Corporation.



Copyright 1994-2014 Compact Data Works, Inc.

All Rights Reserved

BacPac for OpenVMS

Introduction

BacPac performs extremely fast local and remote backups to disk and tape, and also performs disk and file duplication between VMS systems which may or may not be clustered.

BacPac is designed to shorten backup times, reduce backup storage requirements, and provide backup functionality not available with VMS Backup or other backup software.

BacPac may be used for routine daily incremental backups, as well as image mode backups. Remote BacPac may be used to save and restore files across a DECnet network. BacPac Copy may be used to make duplicate copies of tapes for archiving.



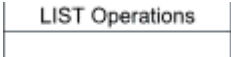
BacPac generates saveset files similar to those generated by the VMS BACKUP Utility. These files are called pacsets. Files restored from a pacset will exactly match the original file in data content and file structure. Files restored during an image mode restore operation will also retain the original file ID.



Using this Help

This Help system contains the entire BacPac Reference Manual. Only the BacPac messages have been omitted - they are available with the BacPac Message Help system.

The help system is complete with an index and glossary and the ability to search for any word. Throughout the help system are many references to other topics in the help system. These will appear as do any links on the Internet. Usually buttons are used for the links. These have special designs as follows;

Button Type	Button example
A product or an operation.	
Qualifier	
Other link	

In addition to navigating around with the buttons, **browse sequences** are provided. The arrows at the top of the Table of Contents will turn red when you enter a browse sequence. You can then use the

arrows to view the next and previous pages in the sequence. For example, when you select an operation from the Table of Contents, you have entered a browse sequence for the operations of that product. For each operation, a browse sequence is provided to take you through all the available qualifiers for that operation.

The **search** works similar to Internet searches except you are only searching in this help system. The **index** is extensive. You can select a topic or type a word or part of a word to view more topics. The 'refresh' or 'Sync TOC' button at the top of the Table of Contents will show you where the current topic is. The red X will 'hide' the Table of Contents. To bring it back, simply click one of the buttons in the help header.

If you are looking for an error message, you will need the **BacPac Messages** help system. Click on the button labeled Messages at the top.

Many examples are presented in the Getting Started chapters as well as with each qualifier. Some knowledge is assumed on the part of the reader with regard to the use of VMS commands and VMS files. The following conventions are used throughout in the examples:

- Square brackets, in command syntax, indicate that a syntactical element is optional. Square brackets are not optional, however, when used to delimit a directory name in a VMS file specification.
- { } Braces surrounding two or more items separated by a comma indicate a choice; you must choose one of the syntactical elements.
- UPPER-CASE In the examples, items shown in upper-case may be entered at the keyboard without any changes.
- lower-case In the examples, items shown in lower-case require a substitution by the user before the command may be typed at a keyboard.



Online Help

Online help is provided with the BacPac installation kit. During installation, it is added to the VMS help library and is accessible with the following VMS command:

\$ HELP BACPAC

The online help is intended as a quick reference to the command syntax and the many qualifiers available. For an experienced user, it may well suffice in place of this help system.



Installation

Installation Requirements

The installation kit consists of one distribution volume labeled VAX BACPAC version k.n-m, AXP BACPAC version k.n-m, or IA64 BACPAC version k.n-m, where k represents the major version number, n represents the minor version number and m represents the revision level.

The installer must have system manager privileges. The installation procedure will take about 5 minutes to run depending on the configuration of your system.

The installer should be familiar with the "Guide to Open VMS Software Installation" and the OpenVMS installation procedure VMSINSTAL.COM.

The version of OpenVMS running must be as follows:

VMS Version 5.2 or higher (VAX)

VMS Version 6.2 or higher (Alpha)

VMS Version 8.2 or higher (Integrity)

Installation Options

BacPac commands may be permanently installed in the DCL tables; the BacPac HELP may be integrated with the VMS HELP. Updating the DCL tables makes BacPac available to all users just as any other VMS utility. Otherwise, the BacPac commands are available only to those users who execute the proper SET COMMAND after logging in.



Selecting an Installation Directory

The installation of BacPac requires one directory. The logical name BP_Manager is defined to point to this directory. Examples of files which reside in this directory are:

- A copy of the BACPAC.CLD file which defines the BacPac command.
- The BacPac HELP if it is not integrated with VMS HELP.
- The BacPac and Remote BacPac executable image files.

The default for the BacPac directory is to create a system level directory called SYS\$SYSROOT:[BACPAC].

During the installation, a list of system files which are new or modified by the installation procedure is displayed. If the BacPac command is integrated with the system DCL tables, users who are logged on during the installation must first log off and then log in again in order to access the BacPac command. If the command is not integrated with the system DCL tables, any user wishing to use BacPac must first execute the following command after logging in:

```
$ SET COMMAND BP_MANAGER:BACPAC
```

This command may also be added to the user's LOGIN.COM command file. Logical names are also set up for the BacPac image and the BacPac help library if the help was not integrated with the system help files. The image logical name is BACPAC.

The logical name for the help library will be HLP\$LIBRARY unless it has already been assigned. The name would then be one of HLP\$LIBRARY_1...HLP\$LIBRARY_n where the name chosen is the first not already in use.

A startup command file is also created by the procedure which defines all the BacPac logical names. It resides in SYS\$MANAGER and should be invoked from the SYSTARTUP_VMS.COM file to insure all logical names are defined after the system is restarted. The following command should be added to the SYSTARTUP_VMS.COM file:

```
$ @SYS$MANAGER:BACPAC_STARTUP
```

This command is automatically executed by the installation procedure.



Conventions used during Installation

The conventions used in the following procedure steps are as follows:

User input is underlined>.

All input and output generated as a result of the installation procedure is in bold type.

The default answer (for those questions that have a default) is specified in brackets ([]) at the end of the question.

The installation procedure can be aborted by pressing <CTRL Y> at any time. In addition, you may get help by entering a question mark (?) at any prompt. The entire process should take about 5 minutes, depending upon your configuration.



Installation Procedure

The following steps are required to install BacPac:

1. Log in to the SYSTEM account.

```
$ Username: SYSTEM
```

```
$ Password:
```

2. Invoke VMSINSTAL at the DCL prompt.

```
$ @SYS$UPDATE:VMSINSTAL
```

3. If there are other users on the system or if you have DECnet running, VMSINSTAL will issue a warning and ask you if you want to continue. It is normally not necessary for other users to log off, and DECnet being active should not cause any problems with the installation.

* Do you want to continue anyway [NO]? Y

4. The next question refers to the backup of your system.

* Are you satisfied with the backup of your system disk [YES]?

5. Next, you will be asked about the distribution volume. Enter the device from which you are loading the installation kit. This may be the name of a tape drive (e.g. MTA0, MFA0).

* Where will the distribution volumes be mounted:

6. Mount the installation volume, enter VAXBACPAC or AXPBACPAC for the product name and confirm that you are ready.

* Products: VAXBACPAC

* Enter installation options you wish to use (none):

Please mount the first volume of the device on [device name]

* Are you ready? Y

7. You will now be asked questions about purging old files replaced by this installation and running the Installation Verification Procedure (IVP) at the end of the installation. It is recommended that you answer YES to both questions.

* Do you want to purge files replaced by this installation [YES]?

* Do you want to run the IVP after the installation [YES]?

8. You will now be asked if you want to create a system level directory in which to store the BacPac files. If you answer YES, the procedure will create the system level directory SYS\$SYSROOT:[BACPAC]. If you answer NO, the procedure will ask you to give the directory specification of a user level directory. If the directory does not exist, the procedure will create it. In all cases, the system logical name BP_MANAGER will be defined to point to the BacPac directory.

* Do you want to create a system level directory [YES]?

* Enter the device and directory for BacPac installation:

9. The next question regards installation of the BacPac command in the system DCL tables. If you answer YES, users will be able to access the BacPac command as other VMS commands. If you answer NO, users will be required to execute a SET COMMAND before they can access BacPac.

* Do you want the BacPac command permanently installed in the DCL tables [YES]?

10. The next question regards installation of the BacPac Server. It must be installed on each node which is to be a server node.

* Do you want the BacPac Server installed on this node [YES]?

11. You will now be asked if you want the BacPac HELP file integrated with the system help files.

* Do you want BacPac HELP permanently installed in the system help library [YES]?

12. You will now be asked if you already have a BacPac license installed. If you are installing an update to BacPac and you already have a license file, answer YES. If you are installing BacPac for the first time, answer NO. If you answer NO, you will then be asked to enter the evaluation kit key provided. Enter the key as provided to you.

* Do you have a BacPac license already installed [NO]?

* Enter the BacPac evaluation kit key: XXXXXX

13. All questions have been asked at this point. The installation will proceed to completion; the verification procedure will be executed, if it had been requested. If the IVP is not successful, a message will be output and the installation terminated.

14. If you did not request to have the IVP executed, you will need to execute the command file BACPAC_STARTUP.COM located in SYS\$MANAGER.

.....

Installing BacPac as a known image

You may want to install BacPac as a known image if there are going to be many users. This will make loading of the program faster. The following command will install BacPac:

```
$ INSTALL
```

```
INSTALL> ADD/OPEN/HEADER BACPAC
```

```
INSTALL> EXIT
```

These commands may be inserted in the BACPAC_STARTUP.COM command file in SYS\$MANAGER or alternatively added to the normal VMS startup command file. If you insert these commands in BACPAC_STARTUP.COM, be sure to insert them at the end of the procedure; this will insure the logical names are properly defined. Refer to the OpenVMS Install Utility Manual for further details.

.....

BacPac License Files

BacPac requires a license file. This is a VMS file called BACPAC.LICENSE. The license file is created by the installation procedure and resides in SYS\$COMMON:[SYSMGR].

The license files created by the installation procedure are of limited time duration. Once the license files expire, BacPac will cease to function.

The expiration date is displayed using the product operation with the /VERSION qualifier. To see when your copy of BacPac will cease to execute, simply type the following command after BacPac is successfully installed:

```
$ BACPAC/VERSION
```

The version of BacPac you currently have will be displayed as well as the date of termination of the evaluation kit.

.....

Updating a License file

To obtain a permanent license file, you will need to obtain the node ID of each system for which you desire a license. To do this, simply run the license procedure BP_LICENSE.COM.

```
$ @BP_MANAGER:BP_LICENSE
```

You should have been provided with a License Registration Form. If you do not have one, call Technical Support to obtain one. Print the node IDs that are generated by the procedure and mail or FAX the printout to Compact Data Works. You will then be provided with a new checksum and instructions on how to upgrade your license file. Once that is accomplished, you have a permanent license file.

If you now display the version of BacPac, an expiration date will no longer appear.

.....

BacPac Command Format

BacPac Command Syntax

The BacPac command is structured in the same manner as all VMS commands. The syntax requires a verb, one or two parameters (input-file-spec and output-file-spec) and optional qualifiers.

Format:

```
$ verb [/qualifier1.../qualifiern] input-file-spec output-file-spec
```

The verb tells DCL which program it is to execute. The verb used to invoke BacPac is BACPAC (or alternatively BACP).



Operations

Save and restore operations are determined from the command line based on one of the file-specs being a tape device. If the input-file-spec is a tape device, it is a restore operation. If the output-file-spec is a tape device, it is a save operation. Disk to disk saves and restores require the /PACSET qualifier to differentiate. A list operation requires the /LIST qualifier and does not use an output-file-spec. A copy operation requires the /COPY qualifier.



Parameters

The input-file-spec is required for save and restore operations and, except for listing a journal file, it is required for list operations. For a save operation, it identifies a disk or set of files to be saved. For a non-image mode save, more than one file specification may be provided, separated by commas. All parts of the second and succeeding file specifications may be omitted. Defaults will be taken from the preceding file specification. This is referred to as sticky defaults. For example, the following two lists of files specify the same set of files:

```
SYSDISK:[TEST]*.DOC,*.ASM
```

```
SYSDISK:[TEST]*.DOC,SYSDISK:[TEST]*.ASM
```

The first example takes advantage of the support for sticky defaults, i.e., all the files will be taken from the SYSDISK device and the directory [TEST]. Since the second set of files (*.ASM) are to be taken from the same place, the device and directory need not be specified.

For a restore or list operation, the input-file-spec identifies the pacset to be used in listing or restoring files.

The output-file-spec identifies a file(s) to be created as the result of the command. In a save operation, it is the file specification of a pacset to be created. In a restore operation, it is the device and/or file specification of the disk or files to be restored.



Qualifiers

Each operation may have qualifiers which further identify the user's requirements to the program. Qualifiers have the form

`/qualifier[=value]`

The slash (/) character is required in all cases. Not all qualifiers take a value. These are simply specified as `/qualifier`.

BacPac qualifiers may be global in scope and affect the entire command or local in scope and affect the file spec they follow. Qualifiers which are global in scope are called command qualifiers; their position on the command line has no effect on the operation. Qualifiers which are local in scope affect the file specification they follow.

Most qualifiers may be negated, i.e., they may take the form `/NOqualifier`. Values are not required with this form of the qualifier. This form is used to override a previous qualifier specification or a default value. This is particularly useful if one is using symbols to invoke BacPac; the symbol may specify a qualifier and, when invoked, that particular qualifier may be turned off using the negated form.

All operations and qualifiers may be abbreviated; the number of required characters is the minimum to make the specified operation or qualifier unique. No more than four characters are required for each operation or qualifier to be unique.



BacPac Operations

Save Operations

BacPac supports save operations for image mode, incremental and differential modes as well as file mode. Image mode saves an entire disk. Incremental and differential save files created or modified since the last image mode save. File mode saves whatever files are given in the input-file-spec. All modes are supported by Remote BacPac as well.

Format

BacPac input-file-spec output-destination [/qualifier1.../qualifiern]

Parameters

input-file-spec

The input-file-spec lists the files to be saved. In an image, incremental or differential save, this simply specifies the disk to be saved. If the input is on a remote system, the node name is included in the file-spec.

output-destination

This may be a tape device, a disk device for disk to disk operations, or a remote system (disk or tape). To specify a remote system, the node name is included.

Qualifiers

Many qualifiers are available for save operations. The only required qualifiers are /INCREMENTAL, /DIFFERENTIAL and /IMAGE for incremental, differential, and image saves.

[/ALIAS] [/ASSIST] [/BACKUP] [/BEFORE] [/BLOCK_SIZE] [/BROADCAST] [/COMMENT] [/COMPRESSION]
[/CONFIRM] [/CREATED] [/DELETE] [/DENSITY] [/DIFFERENTIAL] [/ENCRYPT] [/EXCLUDE]
[/EXPIRED] [/IGNORE] [/IMAGE] [/INCREMENTAL] [/JOURNAL] [/LABEL] [/LIST] [/LOCK_DEVICE]
[/LOG] [/MEDIA_FORMAT] [/MODIFIED] [/NOTIFY] [/PACSET] [/PASSWORD] [/PRIORITY]
[/PROTECTION] [/RECORD] [/RELEASE_TAPE] [/REWIND] [/SINCE] [/STATISTICS] [/TAPE_EXPIRATION]
[/VERIFY]

SAVE topics

- Creating a Pacset
- Creating a Pacset on a Remote Node
- Creating a Pacset on Tape
- Differential Save
- Image Mode Save

- Incremental Save



Restore Operations

BacPac supports restore operations for image mode, incremental and differential modes as well as file mode. Image mode restores and initializes an entire disk. Incremental and differential restore files created or modified since the last image mode save. File mode restores whatever files are given in the input-file-spec. All modes are supported by Remote BacPac as well.

Format

BacPac input-pacset output-file-spec [/qualifier1.../qualifiern]

Parameters

input-pacset

The input-pacset specifies the device and filename of the pacset to be restored. The input-pacset may reside on a tape device, disk, or remote system. If the pacset is on a remote system, the node name is included in with the input-pacset.

output-file-spec

This is the disk where files are to be restored. For image, incremental and differential restores, this is only a disk device. For file mode restores, this specifies the starting pathname where files are to be restored. To specify a remote system, the node name is included.

Qualifiers

Many qualifiers are available for restore operations. The only required qualifiers are /INCREMENTAL and /IMAGE for incremental and image restores.

{ /ASSIST } { /BROADCAST } { /BY_OWNER } { /CBT } { /CONFIRM } { /CONTIGUOUS } { /ENCRYPT } { /EXCLUDE }
 { /IGNORE } { /IMAGE } { /INCREMENTAL } { /INITIALIZE } { /LIST } { /LOG } { /NEW_VERSION } { /NOTIFY }
 { /OVERLAY } { /PACSET } { /PRIORITY } { /RECORD } { /REPLACE } { /REWIND } { /SELECT } { /STATISTICS }
 { /TRUNCATE } { /ZERO_DATA }

RESTORE topics

- Restoring a Pacset
- Restoring Files to a Local Node
- Restoring Files to a Remotel Node
- Differential Restore
- Image Mode Restore

- Incremental Restore



List Operations

BacPac can list pacsets on tape, local disk device or remote disk device or tape.

Format

BacPac/list input-pacset [/qualifier1.../qualifiern]

Parameters

input-pacset

The input-pacset specifies the device and filename of the pacset to be restored. The input-pacset may reside on a tape device, disk, or remote system. If the pacset is on a remote system, the node name is included in with the input-pacset.

Qualifiers

Many qualifiers are available for save operations. The only required qualifier is /LIST. It needs to be the first qualifier used.

[/ASSIST] [/BRIEF] [/ENCRYPT] [/EXCEPTIONS] [/EXCLUDE] [/FULL] [/JOURNAL] [/LIST] [/LOG]
 [/NONE] [/NOTIFY] [/PACSET] [/REWIND] [/SELECT] [/SUMMARY] [/VERIFY]

LIST topics

- Listing a Pacset
- Listing a Remote Pacset
- Listing a Pacset on Multiple Tapes
- Listing a BacPac Journal File



Copy Operations

The BacPac Copy operation may be used to copy tapes written by BacPac. It will exactly copy one or more input tapes to one or more output tapes. The input and output tapes do not have to be the same size nor even the same type.

BacPac creates tapes such that files may be listed or restored starting at the 2nd or later tape. The copied tapes support this as well. Other means of COPYing tapes will not preserve this, e.g., VMS COPY..

Format

BacPac/copy input-tape output-tape [/qualifier1.../qualifiern]

Parameters

input-tape

The input-tape specifies the device on which the tape to be copied resides. The tape must be mounted foreign.

output-tape

The output-tape specifies the device on which the new tape resides. The tape must be mounted foreign.

Qualifiers

The only required qualifier is /COPY. It needs to be the first wualifier used.

[/ASSIST][/COPY][/DENSITY][/IGNORE][/LABEL][/MEDIA_FORMAT][/REWIND][/TAPE_EXPIRATION]

COPY topics

- Copying Tape Pacsetst



Duplicate Operations

BacPac supports duplicate operations for image mode as well as file mode. Image mode copies the input disk and restores and initializes the output disk. File mode copies whatever files are given in the input-file-spec to the output-file-spec. This is a client-server operation and requires a node name on **either** the input-file-spec or the output-file-spec. An optional pacset is created.

Format

BacPac/duplicate [node::]input-file-spec [node::]output-file-spec [/qualifier1.../qualifiern]

Parameters

input-file-spec

The input-file-spec specifies the device and filename of the file(s) to be copied. The input-file-spec may reside on a local disk, or remote system. If the input-file-spec is on a remote system, the node name is included in with the input-file-spec.

output-file-spec

This is the disk where files are to be restored. For image mode this is only a disk device. For file mode restores, this specifies the starting pathname where files are to be restored. To specify a remote system, the node name is included.

Qualifiers

Many qualifiers are available for duplicate operations. The only required qualifier is /DUPLICATE. It needs to be the first qualifier used.

```
[/BACKUP] [/BEFORE] [/BY_OWNER] [/COMPRESSION] [/CREATED] [/DIFFERENTIAL] [/DISMOUNT]
[/DUPLICATE] [/ENCRYPT] [/EXCLUDE] [/EXPIRED] [/IGNORE] [/IMAGE] [/INCREMENTAL]
[/INITIALIZE] [/KEEP] [/LOG] [/MODIFIED] [/MOUNT_FOREIGN] [/NAME] [/NEW_VERSION] [/RECORD]
[/REMOUNT] [/REPLACE] [/SINCE] [/VOLUME_NAME]
```

DUPLICATE topics

- Listing a Pacset
- Listing a Remote Pacset
- Listing a Pacset on Multiple Tapes
- Listing a BacPac Journal File

Support

BacPac provides one support qualifier, /VERSION. This lists the version of BacPac which you are currently using as well as the expiration date if you are running a demo kit.

Format

BacPac/version

Qualifiers

```
[/VERSION]
```


Getting Started

Verifying BacPac is Installed and Available

Once BacPac is installed on your system, you may need to issue a **SET COMMAND** to access the BacPac command; this is dependent on the type of installation selected (See the Installation Instructions). The following command will tell you whether BacPac is available to you, what version is installed, and when your evaluation kit will expire. If you are not running an evaluation kit, no date will be given.

```
$ BACPAC/VERSION
```

If you get a message indicating that BACPAC is an invalid verb, you will need to issue a SET COMMAND as shown below:

```
$ SET COMMAND BP_MANAGER:BACPAC
```

If this command was successful, then retry the VERSION command. If you are still having problems, recheck your installation.



Creating a Pacset

A pacset may be created either on tape or disk. The command and related qualifiers are very similar to those used by VMS BACKUP. You may be able to use the same command procedures you are currently using by simply changing the verb from BACKUP to BACPAC. Qualifiers used by BACKUP which are not supported by BacPac are simply ignored.

To create a pacset on disk, the following command may be used:

```
$ BACPAC *.* DISK$TEST:[BACKUPS]MYDIR/PACSET
```

This command will create a pacset containing all the files in your current default directory. The pacset created will be called MYDIR.PAC; it will reside in the directory [BACKUPS] on the DISK\$TEST device. The default filename extension for any pacset generated by BacPac is .PAC. You may override this simply by explicitly giving an extension on the command line (MYDIR.PAC or MYDIR.BCK).

BacPac provides an alternate syntax which you may find easier to use. You specify the operation you desire (**SAVE**, **RESTORE**, or **LIST**). There is no need for the /PACSET qualifier with this syntax. The above command alternatively could have been:

```
$ BACPAC SAVE *.* DISK$TEST:[BACKUPS]MYDIR
```

If you only wanted to backup the highest version of each file in your directory, you would need to specify this by giving a version number of 0, i.e., the above command would look like the following:

```
$ BACPAC *.*;0 DISK$TEST:[BACKUPS]MYDIR/PACSET or
```

```
$ BACPAC SAVE *.*;0 DISK$TEST:[BACKUPS]MYDIR
```

If you would like to see the name of each file as it is being processed, simply use /LOG.

.....

Creating a Pacset on Tape

The real advantage to using BacPac is seen when writing a pacset to tape. The amount of tape required will be about half. You will not see this advantage unless you are writing multiple tapes as in an image save. You will however see the speed improvement. The following example will create a pacset of the [SYS0] system directory tree on the TEST disk above.

```
$ BACPAC DISK$TEST:[SYS0...]*.* MKA500:SYSBACKUP/LOG
```

Using the alternate syntax, the above command would be:

```
$ BACPAC SAVE DISK$TEST:[SYS0...]*.* MKA500:SYSBACKUP/LOG
```

If you would like a journal file, simply add **/JOURNAL** to the command line. The default journal file is BACPAC.PJL. If it already exists, it will be appended to. You can then keep a running journal of all files backed up. You might want to give the journal file a different name; simply add it to the command as **/JOURNAL=filename**. The default extension is .PJL. The journal file may reside on another node. In this case, use the node name as part of the journal file specification.

You may also want a log file of all messages generated during the save operation. You will get all success messages (i.e., a list of all files successfully backed up) as well as any error messages generated. To get a log file simply add a name to the command as **/LOG=filename**. The default extension is .LOG. If you want a log file and you want to see messages at your terminal, use /MON (MONITOR) in addition.

If you see any error messages you do not readily understand, you may refer to the complete explanations given in the online help or in the Messages Help.

.....

Creating a Pacset on a Remote Node

If you have BacPac and the BacPac Server installed on another node, you may create a pacset on that node by simply adding the node name to the output file specification:

```
$ BACPAC DISK$TEST:[SYS0...]*.* node::MKA500:SYSBACKUP
```

Using the alternate syntax, the above command would be:

```
$ BACPAC SAVE DISK$TEST:[SYS0...]*.* node::MKA500:SYSBACKUP
```

This command will create a pacset of the [SYS0] system directory tree and write it to the tape drive MKA500 located on the node **node::**. See the section on Remote BacPac for more details.

.....

Listing a Pacset

If you would like to see the contents of a pacset, simply use BacPac to list it as follows:

```
$ BACPAC/LIST/REWIND MKA500:SYSBACKUP
```

The alternate syntax for this command would be as follows:

```
$ BACPAC LIST MKA500:SYSBACKUP/REWIND
```

The /PACSET qualifier is optional on a LIST operation with either syntax.

If you forget to mount the tape, BacPac will do it for you. If you do mount the tape, remember that it must be mounted foreign. The /REWIND qualifier is required if you want to start at the beginning of the tape. If you omit it, BacPac will attempt to find the requested pacset starting from the current tape position.

The listing will display the contents of the pacset header as well as each file contained in the pacset. You may also verify each file as it is listed.

```
$ BACPAC/LIST/REWIND/VERIFY MKA500:SYSBACKUP
```

This command requests that each file be expanded in memory as it is processed. CRC checking is done and any errors are reported.

You may use the same operations on a pacset on disk. You do not need to use the /PACSET qualifier for listing a pacset on disk.

```
$ BACPAC/LIST/VERIFY DISK$TEST:MYDIR
```

To list a pacset on a remote node, simply add the node name to the beginning of the device specification, e.g.,

```
$ BACPAC/LIST/VERIFY node::DISK$TEST:MYDIR
```

.....

Listing a Pacset on Multiple Tapes

To list a pacset that spans several tapes, simply start with volume 1 of the tape set and issue the list command. To list only tape 2, simply start with volume 2 and issue the same list command. BacPac knows how to interpret the beginning of each tape in order to start in the middle of a pacset. You may use this technique to restore files also. If the file you are looking for spans both tapes, then you must begin with the first tape.

Once you see the file you are looking for you may use <CTRL-C> to abort the remainder of the list. BacPac will continue reading the tape otherwise.

.....

Creating a BacPac Journal File

Journal files are ongoing journals of which files have been backed up. A new journal file may be created each time you create a pacset or an existing journal file may be appended to. You may want

to create and use a journal file for incremental saves only. This way you know each day which files were backed up. You might create a new journal for each week. In any case, it is easy to do:

```
$ BACPAC/SINCE=BACKUP/JOURNAL=WEEKLY [*...] MKA500:FRIDAY
```

This command will save all files on the current disk which have been modified (or created) since the last backup. The journal file WEEKLY.PJL will be appended to with a list of all the files in the current pacset called FRIDAY.PAC which resides on the tape mounted on MKA500: If the journal file did not exist, it would be created.

If you want to keep the journal file on another node, you may do so. Simply add the node name to the journal file specification:

```
$ BACPAC/SINCE=BACKUP/JOUR=node::device:[dir]WEEKLY [*...]
MKA500:FRIDAY
```

Listing a BacPac Journal File

You may list journal files generated by both BacPac and VMS BACKUP using BacPac. VMS BACKUP cannot, however, read BacPac journal files. To list a journal file, use a command similar to the following:

```
$ BACPAC/LIST/JOUR
```

This will list the default journal file BACPAC.PJL. To list a different journal, specify a filename:

```
$ BACPAC/LIST/JOUR=filename
```

Optionally, specify a complete file specification including node name to list a journal file on another node.

Selecting files to be saved

Files may be selected by filename using any combination of wildcards. Files may also be selected based on any one of the four system dates stored in each file header. Files may also be explicitly excluded. The following example selects files based on the creation date.

```
$ BACPAC *.* /SINCE=TODAY/CREATED MUA0:TODAY
```

The /CREATED qualifier is superfluous since it is the default. The /SINCE qualifier is required. You must specify some range for the system times. You may use /BEFORE instead or you may use a combination of /BEFORE and /SINCE. You may however use only one of the system times. These are specified using /CREATED, /MODIFIED, /EXPIRED, or /BACKUP.

Wild cards may be used for the file name, type and version. You must specify at least one of device, directory, filename, or file type. The default for the file name is * and the default for the file type is *.*.

The version will default to ;* (all versions). If you only want the highest version number of each file, specify ;0.

```
$ BACPAC *.*;0/SINCE=BACKUP MUA0:NEW/PACSET
```

This command will save the highest version of each file in the current default directory which has been modified (or created) since the last backup was done.

This is assuming of course that the last backup was done using the /RECORD qualifier. This qualifier (/RECORD) requests BacPac (or VMS BACKUP) to set the BACKUP date field of each file header. The /PACSET qualifier is optional on this command since the output device is a tape (MUA0).

To perform an incremental backup of an entire disk, use a command similar to the following:

```
$ MOUNT/FOR MKA500:
```

```
$ BACPAC/RECORD/SINCE=BACKUP/REWIND DISK$TEST:[000000...] *.*;*
MKA500:TEST/PACSET
```

This command will examine each file on the DISK\$TEST disk starting at the root directory. Only those files which have been modified (or created) since the last backup will be saved. Each file saved will have the current date and time written into the BACKUP date field of its file header. Note that the *.*;* is unnecessary. The same command could have been:

```
$ BACPAC/RECORD/SINCE=BACKUP/REWIND -
$_ DISK$TEST:[000000...] MKA500:TEST/PACSET
```

A shorthand notation is available with BacPac using the **/INCREMENTAL** qualifier. This qualifier specifies **/RECORD**, **/SINCE=BACKUP**, and also requests the entire disk be backed up. The command then is simplified to

```
$ BACPAC/INCREMENTAL/REWIND DISK$TEST: MKA500:TEST/PACSET
```

The pacset will be called TEST.PAC and will reside as the first pacset on the tape (/REWIND was used). If the tape is mounted, it must be mounted foreign. Tape labels are discussed in the section on tape handling.



Restoring a Pacset

Restoring a pacset is as simple as creating one. Reverse the input-file-spec and the output-file-spec as follows:

```
$ BACPAC MYDIR/PACSET *.*
```

or using the alternate syntax,

```
$ BACPAC RESTORE MYDIR *.*
```

will restore the pacset created on disk in the first example. If the files exist in the current directory, they will not be restored. By default, BacPac will not restore a file with an identical file name to one

that already exists. You may override this condition using /NEW_VERSION, /OVERLAY, or /REPLACE, creating new versions of the files or replacing existing versions with those from the pacset.

You may rename files as they are restored or restore them to a different device and/or directory.

```
$ BACPAC MYDIR/PACSET [OTHERDIR]*.NEW
```

will restore the same files but to the [OTHERDIR] directory. All of the restored files will be given an extension of .NEW.

You may also select (/SELECT) specific files from the pacset and/or exclude (/EXCLUDE) files from being restored. If you are restoring a directory tree, be sure to use the ellipsis when specifying the directory tree.

```
$ BACPAC MYDIRTREE/PACSET [OTHERDIR...]
```

This command will restore the previously saved directory tree to a different directory but will retain the tree structure. If you want to collapse the tree structure, eliminate the ellipsis (...).

Since files are completely restored, including the dates in the file header, you may want to set the backup date to zero to indicate these files have not been backed up. You do this by adding /RECORD=0 to the restore command.

You may also override default file characteristics. The /TRUNCATE qualifier sets the end of file mark at the end of data rather than the allocated space. Use /CONTIGUOUS or /CBT to specify contiguous blocks for the restored file. You may also request that files which were previously marked contiguous be changed (use /NOCONTIGUOUS).



Invoking BacPac from Within Batch Files

You may want to invoke BacPac from within a batch command file. If BacPac was installed on your system using the non-integrated approach, remember to issue the following command before invoking BacPac within your batch file.

```
$ SET COMMAND BP_MANAGER:BACPAC.CLD
```

BacPac can be invoked from within DCL command files which are executed interactively or as batch jobs. As with most VMS utilities, BacPac generates a condition code when it completes execution. If BacPac completes successfully, the condition code will have a true (odd) value. If BacPac completes unsuccessfully, the condition code will have a false (even) value. The condition code can be checked by testing the global symbols \$STATUS or \$SEVERITY. See the "Guide to Using DCL and Command Procedures on OpenVMS" for more information.



Translating an Exit Code

If you are running BacPac from within a batch file or called from another program, there may be times when you wish to translate an exit code. BacPac messages are provided with the distribution kit and

reside in the BP_MANAGER directory. To translate a given exit code, execute the following DCL commands:

```
$ SET MESSAGE BP_MANAGER:BACPAC_MSG
$ symbol = F$MESSAGE(exit_code)
$ SHOW SYMBOL symbol
```

This will display the text of the message associated with the exit code.

.....

Using CTRL-T to Monitor Progress

During any BacPac operation, BacPac will 'piggyback' a status message to the one already provided by VMS with the use of the <CTRL-T> key. The message will indicate which file is being saved or restored and the block number of the file that BacPac is currently processing. This is most useful if you are saving entire disks and would like to see 'progress reports' as the operation proceeds.

BacPac also reports the elapsed time since the start of the operation and a percent done status. The percent done indicates the percent of the files that have been processed, not time. The latter is available only for image mode saves and restores.

.....

Setting the Process Priority Using /PRIORITY

If you are running BacPac interactively, or even from a batch job, you may wish to lower the priority of the process executing BacPac. This is accomplished easily using the /PRIORITY qualifier. The following example will save a file and set the process priority to 3.

```
$ BACPAC/PRI=3 LARGE.DAT MUA0:LARGE
```

The ALTPRI privilege is required for the priority to be raised. If you do not have this privilege, the /PRIORITY qualifier can only be used to lower the priority. Attempts to increase it will have no effect. At the end of the operation, the priority is changed back to what it was before the /PRIORITY qualifier took effect.

.....

Image Mode BacPac

Image Mode Save

The simplest and most important backup operation, and possibly the most efficient, is an image mode save operation. This operation will write the contents of an entire disk to tape (or even another disk). Volume information is saved with the pacset in order to initialize the disk should an image mode restore become necessary. Depending on the size of your system, this type of save operation may be as fast as an incremental backup and certainly makes restoration of files simple. Use the **/IMAGE** qualifier to specify this type of backup. Since the entire disk is saved, file specifications are not allowed.

```
$ BACPAC/IMAGE DISK$USER: MKA500:USERSAVE/REWIND
```

This command will create an image mode pacset of the disk called DISK\$USER. Volume information as well as all files will be saved. The pacset will be called USERSAVE.PAC on the tape mounted in MKA500.

To get around the problem of files open by another user, use the /IGNORE qualifier. This qualifier may also be used to save files which are marked 'Backups Disabled'.

```
$ BACPAC *.* DISK$TEST: [BACKUPS]MYDIR/IGNORE=INTERLOCK
```

This will save files which are currently open by a user or the system. These files may not be restored as desired, i.e., changes may not have been saved. If you want to insure the integrity of your disk, the safest approach is to back it up when no files are open for write access. Use VMS BACKUP to accomplish this. This, however, requires rebooting the system and having it unavailable for some time. An alternative is to use /LOCK which will write lock the disk for the duration of the image save.

This allows users to have read access to the disk during the save, insures that all files backed up are complete, and does not require bringing the system down. **This should not be done on a system disk however.** The system can lock up, requiring a reboot. It has many advantages, however, for a user disk.

If you would like to notify users that an image mode save is about to be done, you can do so with the /BROADCAST qualifier. The broadcast message will be sent to each user terminal and indicate the device being backed up. It will also tell them if the device is being locked. They may then decide to suspend write operations until the backup is complete (if you have not specified /LOCK). As soon as BacPac is finished writing the pacset, before the record pass if any, it will send another broadcast message informing users that the backup is complete and the disk is unlocked (if it was in fact locked).

If you normally run image mode saves from a batch file and would like to know the status as the job completes, you may use the /NOTIFY qualifier which will send you mail (or another user if you so specify) when BacPac is done. If errors occurred, BacPac will mail you the error log (PACERROR.LOG). If no errors occurred, BacPac will simply tell you this. You may use the /NOTIFY qualifier with any BacPac operation.



Image Mode Restore

While an image mode save operation may be run with the disk to be saved mounted and in use by others, an image mode restore operation requires total access to the disk. The disk to be restored must be mounted foreign. BacPac will initialize the disk and then restore all files from the image mode pacset. If the disk to be restored is the system disk,

To restore a disk, simply use a command such as the following:

```
$ DISMOUNT DKA0:  
$ MOUNT/FOREIGN DKA0:  
$ BACPAC/IMAGE/REWIND MKA500:DK0SAVE DKA0:
```

This set of commands will completely restore the disk DKA0: from the pacset DK0SAVE.PAC on the tape mounted on MKA500:. The parameters used to initialize the disk will be taken from those saved during the image mode save. If you wish to change the initialization parameters, e.g., the number of file headers on the disk or the cluster factor, you may initialize the disk before running BacPac.

```
$ DISMOUNT DKA0:  
$ INITIALIZE/HEADERS=10028 DKA0: HENRY  
$ MOUNT/FOREIGN DKA0:  
$ BACPAC/IMAGE/REWIND/NOINITIALIZE MKA500:DK0SAVE DKA0:
```

These commands will first initialize the disk, give it a volume name of HENRY and create 10028 file headers. BacPac will then use this volume name when it restores the disk rather than whatever the disk was previously called. It will check the number of file headers to insure that it can restore all the files. If the number of file headers is insufficient BacPac will increase this number to allow restoration of all files. It will never decrease it.

An image mode restore is quite useful for changing parameters on a disk, such as the number of file headers or the maximum number of files allowed, and has the added benefit of a completely defragmented disk.

Since BacPac requires the disk to be mounted foreign, it is pretty safe that you will not accidentally restore the wrong disk. However, you have the option to see exactly what BacPac intends to do before it does it. Use the /CONFIRM qualifier and BacPac will tell you what volume and device it is about to restore, and from what tape and pacset mounted on which device. You then have the option to abort the process or continue. A sample prompt follows:

```
%BACPAC-I-INIT_DISK, Volume <volume> on <device> will be initialized and  
restored from Pacset <Pacset> on tape volume <label> on <device>
```

```
Enter "YES" to continue or "QUIT" to abort:
```

/CONFIRM provides a similar prompt for image mode saves.

.....

Incremental and Differential Backups

Incremental Save

An incremental save differs from an image save in that only files which have been created or modified since the last backup will be saved. Since image mode save operations may be too lengthy to be performed on a daily basis, you will probably perform incremental saves more regularly. An incremental save is accomplished using the /SINCE=BACKUP and /RECORD qualifiers.

```
$ BACPAC/SINCE=BACKUP/RECORD DISK$USER:[000000...]*.*;*-  
$_ MKA500:DATE
```

This command will save all files on the disk DISK\$USER which have not been backed up since the last pacset was made. The current date will be recorded in the backup date field of each file which is saved. You need to use /RECORD in order to have /SINCE=BACKUP be effective. Each successive incremental save will save only those files which have been created or modified since the last incremental save.

BacPac provides a shortcut for the incremental operation, i.e., to accomplish the above example, the command would simply be:

```
$ BACPAC/INCREMENTAL DISK$USER: MKA500:date
```

No file specification is given on the command line; it is analogous to using /IMAGE. /INCREMENTAL implies /SINCE=BACKUP and /RECORD.

All VMS system disks contain alias files. User disks may also contain alias files. This is a file whose name appears in more than one directory. The file itself resides on the disk only once. If you are saving an entire disk as part of an incremental save, you may actually save alias files 2, 3 or more times. They will get saved once for each appearance in a directory. This is true for both BacPac and BACKUP. **If you do not want these files to be saved more than once, use /NOALIAS on the command line.** Each file will be saved once, at the first occurrence encountered. This is taken care of automatically if you use the /INCREMENTAL qualifier.



Differential Save

An incremental save is the fastest save operation between image save operations, except that it must do a record pass. However, if you are ever required to restore a disk, the incremental restore operation is not straight forward. An alternative is to perform differential saves between image save operations. The operation is identical except that the /RECORD qualifier is not present. Each differential save thus saves all files modified or created since the last image save. This actually saves more data than the incremental save but does not require a record pass. The restore operation requires only the image restore followed by one differential restore.

BacPac provides a shortcut for the differential operation, i.e., to accomplish the above example, the command would simply be:

```
$ BACPAC/DIFFERENTIAL DISK$USER: MKA500:date
```

This command will not record the backup date for each file.



Incremental Restore

You may want to keep a journal of the incremental saves done between each image mode save. This makes retrieving an individual file easy; you can tell rather quickly from the journal file when the particular file was backed up. To keep a journal simply add /JOURNAL or /JOURNAL=filename to the command line. BacPac will append to the journal file if it already exists. After each image save, simply use a new journal file name or delete the old one. If you do not specify a journal name, BacPac will default to BACPAC.PJL.

If the catastrophic happens, your disk crashes etc., you may now restore your disk to the state of the last incremental save. If all of your backups were image mode saves, you would only need restore the one pacset. In any event, this is the first step in restoring your disk. For this step, you will use the /IMAGE and /RECORD qualifiers. Then you restore each of the incremental pacsets to bring your disk to the state it was in at the time of the last incremental save. For this operation, you use the /INCREMENTAL qualifier.

```
$ MOUNT/FOREIGN DUA0:
$ BACPAC/IMAGE=RESTORE/REWIND/RECORD MKA400: DUA0:
$ DISMOUNT DUA0:
$ MOUNT DUA0: diskname
$ BACPAC/INCREMENTAL MKA400:update1 DUA0:
$ BACPAC/INCREMENTAL MKA400:update2 DUA0:
$ BACPAC/INCREMENTAL MKA400:update3 DUA0:
```



Differential Restore

A differential restore differs from an incremental restore in that only one differential save need be applied after the image restore. This is a much faster operation.

```
$ MOUNT/FOREIGN DUA0:
$ BACPAC/IMAGE=RESTORE/REWIND/RECORD MKA400: DUA0:
$ BACPAC MKA400:update3 DUA0:
```

Note that no special qualifiers are needed and only the most recent differential save is used.



Remote BacPac

Remote BacPac

BacPac's client/server architecture and network communications provide for saving and restoring files across a DECnet network. BacPac must be installed on each of the DECnet nodes. To invoke Remote BacPac, simply specify a node name along with the device specification. The node name can be specified for the input device or output device, but not both on the same command. BacPac will establish a DECnet connection with the remote node which starts up the BacPac Server.



Saving files to a remote node

One of the major advantages of using BacPac is its built-in compression. The smaller pacsets result in reduced network traffic. Files are compressed on the node where they reside, sent across the network in compressed format, and the pacset is created on the receiving node. For example, to save files from a local node A to a remote tape on node B, the command would look similar to

```
$ BACPAC [TEST]*.* NODE_B::MKA500:TESTFILES/PACSET/REWIND
```

In the above example, files are saved from Node A, sent across the network, and written to tape on Node B. File compression is done on Node A. The BacPac Server will automatically be started on Node B, receive the data from the network and write the tape.



Saving files from a remote node

The BacPac executable image and that of the BacPac Server are virtually mirror images of each other. The BacPac Server, which always runs on the remote system, may either receive files from the network or save files to the network.

```
$ BACPAC nodeb::[000000]*.*;* MKB400:NODEB/REWIND/LOG
```

This command will save all the files on Node B and write the pacset to the local tape MKB400. In this case, file compression will occur on Node B. Files are transferred across the network in compressed format. Log messages are generated on Node B, where the files reside, and also written across the network for display on the local terminal. The same is also true of any error messages generated by the BacPac Server. All messages will appear as if they were generated by the local BacPac except they will be prefixed with BACPACSRV instead of BACPAC. This way, you can tell which messages were generated by the server software and which were generated by the client software.

If you desire a listing file, you will need to specify the node name of the listing file on the command line. The default is to write the listing file to the node where the files reside, i.e., Node B. To create a listing file and have it reside on Node A, the following command could be used:

```
$ BACPAC nodeb::[000000]*.*;* -  
$_ MKB400:NODEB/REWIND/LIS=nodea::disk$user:[logfiles]nodeb
```

This command will create the listing file and write it to the local node in the directory LOGFILES on device DISK\$USER.



Listing a remote Pacset

To list a remote pacset, simply add the node name to the filespec to be listed.

```
$ BACPAC/LIST/REWIND NODEB::MKA500:
```

The pacset will be listed in the same manner as a local tape. In order to list the second pacset on the tape, it would however be necessary to list it using the pacset name. The BacPac Server is logged out at the end of each command; the tape is dismounted but not unloaded. Each new invocation of the BacPac Server will begin at the beginning of tape. If you are not sure which pacsets are on the tape, you may list them all in an abbreviated format by using /SUMMARY. The BacPac Server will scan the tape and return the beginning of the listing of each pacset, i.e., you will see the pacset name, date, command line used, etc. but you will not see any file names.

```
$ BACPAC/REWIND/LIST/SUMMARY NODEB::MKA500:
```

If you already know which pacset you want, simply specify it on the command line. BacPac will issue a message for each pacset that it skips which searching for the one you want.

```
$ BACPAC/REWIND/LIST NODEB::MKA500:TESTFILES
```

The /REWIND is actually redundant in these cases, since the BacPac Server always begins at the beginning of tape.



Restoring files to a local node

Restoring files to a local node from a remote pacset is virtually the same as restoring files from a local tape. Simply add the node name to the file specification of the pacset to be restored.

```
$ BACPAC NODEB::MKA500:TESTFILES/PACSET []/LOG
```

The pacset TESTFILES will be restored to the current default directory on the local system. Files which are compressed are sent across the network in compressed format and expanded on the node where they will be restored, in this case, the local node. Log messages and listings are generated on the node where the files are restored. In this case, except for the tape actually being remote, all actions are the same as a non-remote BacPac.



Restoring files to a remote node

In this case, the node name is placed on the file specification of the files to be restored and the tape is local.


```
$ BACPAC MKB400:TESTFILES/REWIND/PACSET -  
$_ NODEB::DISK$USER:[TEST]/LOG
```

Files are sent to the remote node, expanded there and restored. All LOG and LIST messages are generated on the remote system and sent back to the client system for display. In this case all messages are prefixed with BACPACSRV rather than BACPAC. Since the tape is local, the /REWIND becomes important. Again, the /PACSET qualifier is required.

.....

Image, Incremental and Differential modes

Image mode, incremental mode and differential mode are all supported by Remote BacPac. All other qualifiers are also supported. The only real restriction to using Remote BacPac is the required use of the /PACSET qualifier.

.....

BacPac Duplication

BacPac Duplication

BacPac provides for file and disk duplication using the /DUPLICATE qualifier. Duplication is the logical equivalent of both a save and restore operation without the intervening pacset. In duplicate mode, no pacset is created (unless specifically requested using /KEEP). Files are compressed and copied to the destination where they are expanded and restored. Since this is a network operation, file compression is used to minimize the amount of traffic on the network.

An entire disk may be duplicated using the /IMAGE qualifier. The target disk must be mounted foreign and will be initialized and restored to match the source disk. Several qualifiers are available to assist in mounting the target disk.

Bacpac provides tremendous power and flexibility when duplicating VMS disks from one system to another. Qualifiers allow you to specify how to handle situations such as:

- **Mount the target disk if it is not mounted**
- **Dismount the target disk if it is already mounted**
- **Leave the target disk mounted after the duplication completes**
- **Give the target disk a different volume name**
- **Keep a pacset backup in addition to creating a duplicate disk**



Copying Files to/from a Remote Node

Copying files from one node to another is quite simple. Just specify the source file specification and the destination file specification. One, and only one, of them must have a node name as part of the file specification. While on node a, you can copy files from node a to node b or from node b to node a. You cannot copy files from node b to node c. You can, however, copy files from node a to node a by specifying the node name on either the source file specification or the destination file specification.

Assume you are logged onto node ALPHA. The following command will copy all the files in your current directory to the [DATFILES] directory on device DISK\$USER on node BETA.

```
ALPHA $ BACPAC/DUPLICATE *.* BETA::DISK$USER:[DATFILES]*.*
```

To copy files from node BETA to node ALPHA :

```
ALPHA $ BACPAC/DUPLICATE BETA::DISK$USER:[DATFILES]*.* *.*
```

This command will copy all the files in the [DATFILES] directory on device DISK\$USER on node BETA to your current directory on node ALPHA.

You may exclude some files from being copied and even rename the results in a single command as follows:

```
ALPHA $ BACPAC/DUP *.* /EXCLUDE=*.DAT
        BETA::DISK$USER: [DATFILES] *.NEW
```

All the files in your current directory, except those with file type .DAT will be copied. The newly created files will all have file type .NEW.

All of the file selection qualifiers available for BacPac are also available when duplicating files.

To copy a file to the same node, simply put the node name on either the source or the destination specification.

```
ALPHA $ BACPAC/DUPLICATE *.* ALPHA::DKA100: [DATFILES] *.*
```



Duplicating a local disk to/from a remote node

The target disk of an image mode duplication must be mounted foreign. If the target disk is mounted, it must first be dismounted and then mounted foreign. BacPac provides /DISMOUNT (to dismount the target disk) and /MOUNT_FOREIGN (to then mount the target disk foreign). Sufficient user privilege is required for all these operations.

Following are examples of how to duplicate a local disk on local node alpha to a remote node (node bravo):

- Duplicate a local disk on node alpha to a remote target disk on node bravo. In this case the target disk is not mounted. BacPac will mount the target disk foreign.

```
alpha $ BACPAC/DUP/IMAGE DISK$USER1: BRAVO:: $3$DKA0: /MOUNT_FOREIGN
```

- Duplicate a local disk on node alpha to a remote target disk on node bravo. BacPac will dismount the target disk if it is mounted and mount it foreign before the operation commences.

```
alpha $ BACPAC/DUP/IMAGE DISK$USER1: BRAVO:: $3$DKA0: /DISMOUNT/MOUNT
```

- Duplicate a local disk on node alpha to a remote target disk on node bravo. BacPac will dismount the target disk if it is mounted and mount it foreign before the operation commences. BacPac will remount the target disk for cluster and system wide after the duplication is complete:

```
alpha $ BACPAC/DUP/IMAGE DISK$USER1:
BRAVO:: $3$DKA0: /DISMOU/MOU/REMOUNT
```

- Duplicate a remote disk on node bravo to a local target disk on node alpha. BacPac will dismount the target disk if it is mounted and mount it foreign before the operation commences. BacPac will remount the target disk for cluster and system wide after the duplication is complete:

```
alpha $ BACPAC/DUP/IMAGE BRAVO::DISK$USER2:
$4$DKA100:/DISMOU/MOU/REMOU
```

.....

Mounting the Target Disk

BacPac normally leaves the target disk of an image mode operation unmounted. This is true after a pacset restore as well as an image mode duplication. This may not be the desired result of a duplication. BacPac can mount the target disk for cluster and system wide use after an image mode duplication (/REMOUNT). This may have undesirable results if the target disk is part of the same cluster as the source disk. Boths disk would then have the same volume name and both could not be mounted at the same time. BacPac provides /VOLUME_NAME to rename the target disk so it may be mounted. The two disks are virtual copies of each other except for the volume name.

BacPac also provides for specifying the logical name (/NAME) to be assigned to the target disk when it is remounted.

The following command will duplicate a remote disk on node bravo to a local target disk on node alpha. The target disk will be dismounted and mounted foreign before the operation commences. At completion, the target disk will be mounted cluster wide and given a different volume name and logical name.

```
alpha $ BACPAC/DUP/IMAGE BRAVO::DISK$USER2: $4$DKA100:/DISMOU
/MOUNT /REMOUNT/VOLUME_NAME=USER2DUP/NAME=DISK$DUP
```

.....

Duplication and Incremental Backups

Once an image mode duplication has been completed, it may be desirable to keep the target disk up-to-date on a regular basis. The easiest and fastest way to do this is to use /INCREMENTAL. This would necessitate using /RECORD on the initial image mode duplication to set the backup dates of all the source files on the source disk. The incremental duplication only copies the files that have been created or modified since the last duplication. /INCREMENTAL automatically specifies /SINCE=BACKUP.

First complete the image mode duplication with /RECORD.

```
alpha $ BACPAC/DUP/IMAGE/RECORD DISK$USER1: BRAVO::$3$DKA0:
```

Periodically the following command should be issued to keep the target disk up-to-date. This command will scan the entire source disk for files which have been created or modified since the last backup. /INCREMENTAL will also automatically do a record pass at the end (/RECORD) updating the backup dates of all newly copied files.

```
alpha $ BACPAC/DUP/INCREMENTAL DISK$USER1: BRAVO::$3$DKA0:
```

.....

Using Compression

By default, compression is turned on when duplicating files or disks. This minimizes the amount of traffic on the network. Files are compressed on the source node and expanded on the destination node. If you have a slow network link, you may want to try /COMPRESSION=MAX. This will make the amount of network traffic even smaller. It will however increase the amount of CPU time on both the source and destination nodes.

Assume you are logged onto node ALPHA. The following command will copy all the files in your current directory to the [DATFILES] directory on device DISK\$USER on node BETA.

```
ALPHA$ BACPAC/DUP/COMP=MAX *.* BETA::DISK$USER:[DATFILES]*.*
```

.....

Encryption

Encryption Keys and Algorithms

BacPac supports 6 AES encryption algorithms. These vary as to key length and are all stream-mode ciphers. VMS stores the keys and creates a logical name for them. The default is to store the logical name in the process name table. If you want other users to be able to access the key, you will specify a logical name table when you create the key. The algorithms supported by BacPac are:

AESCFB128 - cipher feedback (default)

AESCFB192 - cipher feedback

AESCFB256 - cipher feedback

AESOFB128 - output feedback

AESCFB192 - output feedback

AESCFB256 - output feedback

The key length is critical to each algorithm. The minimum key length required for a 128 bit algorithm is 16; the minimum for a 192 bit algorithm is 24 and for a 256 bit algorithm is 32. If the key is not long enough for the algorithm specified, BacPac will generate an error message and quit.

To generate a system wide key for a 128 bit algorithm, use something similar to:

```
$ ENCRYPT /CREATE_KEY /SYSTEM keyname "this is a sample of an encryption key"
```

The key keyname may then be used in BacPac commands.

.....

Using Encryption

BacPac supports AES encryption through the use of the qualifier /ENCRYPT. This applies to both pacsets and files copied using /DUPLICATE.

To create a pacset using encryption, the following command may be used:

```
$ BACPAC *.* DISK$TEST: [BACKUPS] MYDIR/PACSET/ENCRYPT= (NAME=keyname)
```

This command will create a pacset containing all the files in your current default directory. The pacset created will be called MYDIR.PAC; it will reside in the directory [BACKUPS] on the DISK\$TEST device. The key used for encryption has been previously defined using the VMS ENCRYPT command.

Internally, BacPac uses the supplied key to encrypt a canned message. It stores the checksum of the encrypted message in the pacset. The key (nor its name) is not stored anywhere. To restore the pacset you must use the same key. It may have a different key name but the value must match. If you forget the key, you will never be able to list or restore the pacset. If you are restoring the pacset on a remote system, you will need to define the same key there also.

To restore the above pacset, the following command may be used:

```
$ BACPAC DISK$TEST: [BACKUPS]MYDIR/PACSET/ENCRYPT=(NAME=keyname) *.*
```

If you do not specify a key on the command, BacPac will prompt you for a key. The key will NOT be echoed as you type it. BacPac will ask you to confirm the key. Remember to specify enough characters to satisfy the algorithm you are using. With the following command, BacPac will prompt you for the key value and use the AESCFB128 algorithm. You will need to enter at least 16 characters.

```
$ BACPAC DISK$TEST: [BACKUPS]MYDIR/PACSET/ENCRYPT *.*
```

In order to use encryption with /DUPLICATE, you will first need to define the key on both systems. Duplicate is equivalent to performing a SAVE operation on one system and a RESTORE on the other. If the same key is not available on both systems, the RESTORE operation will fail immediately once the remote server checks the canned message. Both the client and the server 'share' the same command line.

```
alpha $ BACPAC/DUP/IMAGE/ENCRYPT=(NAME=keyname) DISK$USER1: -  
alpha_ $ BRAVO:: $3$DKA0:/MOUNT_FOREIGN
```

Using the above command, BacPac will duplicate the local disk on node alpha to a remote target disk on node bravo. The encryption key used has been previously define on both systems using the VMS ENCRYPT command.

.....

Tape Handling

Tape Labels

BacPac writes ANSI labeled tapes. The first block on a tape describes the label and the format of the tape. The exact layout of a tape, including the label fields, may be found in "Guide to OpenVMS Disk and Magnetic Tape Operations". The tape label is 6 ASCII characters and must be specified when writing to a tape.

You can only write a tape label when you are initializing a tape, i.e., you must specify /REWIND. When you are writing a tape, the /REWIND qualifier requests that the tape be initialized, i.e., a new label written to the tape. When you are reading a tape, /REWIND simply means to start at the beginning of tape. BacPac will supply a default label for you if you specify /REWIND and do not specify /LABEL.

```
$ BACPAC/REWIND *.* MUA0:MYFILES
```

This command requests that BacPac initialize the tape, provide a default label and create a pacset containing the files in the current default directory. The default label will be the 6 characters BACPAC. If the tape is a virgin tape (i.e., never been written to), this command is fine.

If the tape already has a label, BacPac will check the label and the protection fields on the tape. If the label does not match the default BacPac label, the expiration date has not been reached, the accessibility field is not blank, or the user does not have sufficient privilege to write to this tape, BacPac will issue an OPCOM message to the Tapes Operator requesting that the proper tape be mounted. You can override this action by adding the qualifier /IGNORE=(LABEL_PROCESSING,ACCESSIBILITY). This qualifier requests BacPac to ignore all of the above checking and write the tape as if it were a virgin tape.

The safer approach is to supply BacPac with the correct label and let BacPac check the other fields. The above command could have been the following;

```
$ BACPAC/REWIND/LABEL=MYTAPE *.* MUA0:MYFILES
```

With this command, BacPac will verify that the label on the tape is 'MYTAPE', other verification will also be done, and the tape will be initialized with the new label 'MYTAPE'. This prevents accidentally overwriting a tape which was incorrectly mounted.

Remember that the label applies to the entire tape, not just the first pacset written to it. You may want to write several pacsets to the same tape, so the label should be descriptive of what the entire tape contains.



Default Tape Labels

BacPac uses the default tape label BACPAC. VMS BACKUP uses the pacset name for the default tape label. You may prefer this latter approach. In this case, simply define the logical name **BACPAC_LABEL** and give it an equivalence name of PACSET. BacPac will then use the pacset name as the default label in the same manner as BACKUP. You may also set the equivalence string to be DATE. In this case, BacPac will generate a label in the form MMDDYY. This is handy if you are going to write several pacsets to the same tape.

```
$ DEFINE BACPAC_LABEL PACSET  
$ BACPAC/REWIND [*...]*.* MUA0:USERFILES
```

In this example, the pacset name will be USERFILES.PAC. The tape label will be taken from the first 6 characters of the pacset name, hence the label written to the tape will be USERFI.

```
$ DEFINE BACPAC_LABEL DATE  
$ BACPAC/REWIND/IMAGE DISK1: MKA300:DISK1_IMG
```

In this example, the pacset name will be DISK1_IMG.PAC and the label will be the current date in the form MMDDYY.



Tape Security

When it comes to tapes, the best security is physical security. VMS offers limited software security for tapes. The combination of software and physical security should suffice except for the accidental overwrite of a tape. You should write protect all tapes you do not want overwritten and keep them in a safe place.

BacPac performs all security checking that is available. First, it will check the tape label to be sure it is what the user intended. This is true for reading tapes as well as writing them. It also checks the accessibility code. If it is anything other than a blank, it will not read or write the tape. Of course, this can be overridden using the /IGNORE=ACCESSIBILITY qualifier. BacPac also checks the protection field against the UIC of the user to insure that the user has write (or read) access to the tape or has sufficient privilege to override this. Using /IGNORE=LABEL_PROCESSING removes all tape checking. This option should be used with great care.

Tapes may be written with an expiration date. This is another check that is performed before allowing initialization of the tape. If the tape expiration date has not yet passed, you will not be allowed to initialize the tape. In all of these cases, an OPCOM message will be sent to the Tapes Operator requesting mounting of the proper tape.

BacPac also provides password protection to pacsets. This can prevent someone from reading data you do not want them to have access to, but does not prevent overwriting a tape. Use the /PASSWORD qualifier to specify this. When using /PASSWORD without a value, BacPac will prompt you for the password. This way you can provide one without having to type it on a command line for others to see. If you use /PASSWORD, remember, there is no way to recover if you forget it. BacPac does not store the password as a string and there is no override feature for a forgotten password.



OPCOM Messages

BacPac will not proceed to initialize or write a tape until it has verified the label and performed the other protection checks, if not overridden. If any checks do not succeed, an OPCOM message will be generated. Other OPCOMs include mounting of a second volume of a tape.

If you do not want OPCOM message generated, you may use /NOASSIST. BacPac will generate prompts which are sent to your terminal instead. Simply answer the prompt, and BacPac will process your response and either continue or abort processing.

If you select /ASSIST and there is no operator enabled to answer your request, you will receive a message indicating such and a prompt as if /NOASSIST had been used.



Multi Volume Tape Pacsets

If you are saving an entire disk, it is quite possible that the pacset will span multiple tape volumes. BacPac will issue an OPCOM (or prompt if you used /NOASSIST) to mount the second and subsequent tape volumes. Tape labeling is automatic, after label checking. The tape label will match that of the first volume with the last 2 characters replaced by the characters 'nn' where nn is the volume number of the tape, unless you have provided a list of labels. The list will be used in the order given until it is exhausted. If additional tapes are required, the first label supplied is used with the tape volume number in character position 5 and 6. If any of the labels are shorter than 4 characters, the blank characters are replaced with underscore characters when adding the volume number. For example, if the label supplied is 'A', the label written to the second volume would be 'A__02'.

Multi volume pacsets may also be written to a list of devices:

```
$ BACPAC/REWIND DISK1: [*...] MUA0:DISKSAVE,MKA300:
```

In this case, the pacset is written to the tape in MUA0. When that tape fills up, BacPac automatically switches to the tape in MKA300. Tape labeling applies as described above. There may be any number of devices supplied. In the case above, if the tape in MKA300 fills up, BacPac will switch back to the tape in MUA0. The operator will be prompted to mount another physical tape since the first tape will have been unloaded.



Copying Tape Pacsets

It is often desirable to have a copy of a tape pacset to store at another location or to send to another site. BacPac provides a copy capability for this purpose. To copy a tape mounted in MUA0: to a tape mounted in MKA600:, simply issue the following command:

```
$ BACPAC/COPY MUA0: MKA500:
```

BacPac will copy the tape in its entirety, including subsequent volumes. This is preferable to using VMS COPY for several reasons. With VMS COPY, you must mount both tapes structured prior to issuing the COPY command. VMS COPY will then write tape label records which do not match the original ones written by BacPac. You may or may not be able to read volumes 2-n of the copied set. You will not be able to read this tape set starting at any volume other than one.

BacPac Copy is the proper way to copy a BacPac tape. The tapes are both mounted foreign and a logical copy is made. An exact copy would require tapes of the same size on like drives. This is not necessary. BacPac Copy will copy the tape to any other tape device or to disk. You may also copy disk pacsets to tape with the proper BacPac label and information for reading the second and subsequent tapes.

While no qualifiers are necessary for copying a tape set, you may provide /LABEL and /IGNORE=LABEL_PROCESSING. BacPac Copy checks the tape label on the output tape the same as BacPac. Since it is likely that the output tape is an archival tape, you may change the tape expiration date if you like by specifying /TAPE_EXPIRATION.



Disaster Recovery License

Disaster Recovery License

The executable image `BACPAC_RESTORE.EXE` is supplied to those customers who may need to use BacPac on a system other than their licensed system for purposes of disaster recovery. This image is simply copied to the disaster recovery system and is invoked using a foreign command.

```
$ RESTORE := $DISK$DUA0: [BACPAC] BACPAC_RESTORE.EXE
```

Invoke the foreign command to list or restore the contents of pacsets.

To list the contents of a BacPac pacset:

```
$ RESTORE BACPAC/LIST MKA600:DISK1/PAC
```

To restore the contents of a BacPac pacset:

```
$ RESTORE BACPAC/IMAGE MKA600:DISK1/PAC DUA1:
```

.....

Qualifiers

Qualifiers Reference List

Each of the available BacPac qualifiers is discussed in this section. They are presented in alphabetical order. Each qualifier is presented with a detailed description of its function, applicable operations, and its format.

[/ALIAS] [/ASSIST] [/BACKUP] [/BEFORE] [/BLOCK_SIZE] [/BRIEF] [/BROADCAST] [/BY_OWNER]
[/CBT] [/COMMENT] [/COMPRESSION] [/CONFIRM] [/CONTIGUOUS] [/COPY] [/CREATED]
[/DELETE] [/DENSITY] [/DIFFERENTIAL] [/DISMOUNT] [/DUPLICATE] [/ENCRYPT] [/EXCEPTIONS]
[/EXCLUDE] [/EXPIRED] [/FULL]
[/IGNORE] [/IMAGE] [/INCREMENTAL] [/INITIALIZE] [/JOURNAL] [/KEEP] [/LABEL] [/LIST]
[/LOCK_DEVICE] [/LOG] [/MEDIA_FORMAT] [/MODIFIED] [/MOUNT_FOREIGN] [/NAME] [/NEW_VERSION]
[/NONE] [/NOTIFY] [/OVERLAY]
[/PACSET] [/PASSWORD] [/PRIORITY] [/PROTECTION] [/RECORD] [/RELEASE_TAPE] [/REMOUNT]
[/REPLACE] [/REWIND] [/SELECT] [/SINCE] [/STATISTICS] [/SUMMARY] [/TAPE_EXPIRATION]
[/TRUNCATE] [/VERIFY] [/VERSION] [/VOLUME_NAME] [/ZERO_DATA]

/ALIAS

Requests file processing similar to image mode.

type

Command qualifier.

format

/ALIAS (default)

/NOALIAS

operations

SAVE
<div style="background-color: green; width: 50px; height: 10px;"></div>

related qualifiers

none

restrictions

not available for image mode

description

Alias processing is the normal mode of incremental save processing. Aliased files are saved intact if they are encountered in the input-file-spec. Files may actually be saved twice. Use of /NOALIAS inhibits processing of alias files. These files are only saved when encountered in the directory in which their header points. /NOALIAS is the normal mode of image mode saves. Files are only saved once. The drawback occurs if you want to restore an alias file using its alias name.

example

```
$ BACPAC/NOALIAS/SINCE=TODAY [000000...]*.*-  
$_MKA500:DIRTREE/LABEL=TODAY (1)
```

(1) All files on the current disk which have been created today will be saved in the pacset DIRTREE.PAC which will reside on the tape labeled TODAY on device MKA500. Any files encountered on the disk which are aliased will not have their data saved. Only the header file is saved. Files saved in this manner may only be retrieved using the file specification which contains the directory of their file header.



/ASSIST

Requests operator assistance for mounting tapes.

type

Command qualifier.

format

/ASSIST (default)

/NOASSIST

operations



related qualifiers

none

restrictions

none

description

This qualifier provides for operator assistance with tape handling. An OPCOM message will be sent to the Tapes Operator whenever a tape needs to be mounted. If you select **/NOASSIST**, a prompt is sent to the terminal or window where the BacPac command was issued whenever a tape needs to be mounted. Messages will also be issued if the incorrect tape is mounted, e.g., the tape label does not match the one requested.

If there is no operator enabled to handle requests, you will receive a message indicating such and a prompt as if **/NOASSIST** had been requested.

example

```
$ BACPAC/ASSIST/REWIND/LABEL=MYTAPE DISK$TEST:[000000...]*.*
MKA600:MYTAPE (1)
```

(1) A message will be sent to the Tapes operator to mount the initial tape labeled MYTAPE on MKA600:. If the pacset requires a second or subsequent tapes, additional messages will be sent to the Tapes Operator.

All files on the disk DISK\$TEST will be saved to the pacset MYTAPE.PAC which will be written to tape in MKA600:.

```
$ BACPAC/NOASSIST/REWIND/LABEL=MYTAPE DISK$TEST:[000000...]*.*
MKA600:MYTAPE (2)
```

(2) A prompt will be issued to mount the initial tape labeled MYTAPE on MKA600:. If the pacset requires a second or subsequent tapes, additional prompts will be issued. The pacset will be the same as in (1).

.....

/BACKUP

Requests that files be selected based on the date the file was last backed up.

type

Input file selection qualifier.

format

/BACKUP/BEFORE=time

/BACKUP/SINCE=time

/NOBACKUP

operations

SAVE	DUPLICATE
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

related qualifiers

[/BEFORE] [/CREATED] [/EXPIRED] [/IGNORE] [/MODIFIED] [/RECORD] [/SINCE]

restrictions

requires **/BEFORE** or **/SINCE**

description

/BACKUP causes files to be selected from the input-file-spec based upon the system time recorded in the backup date field of each file's header record. This time is displayed by VMS during execution of a DIRECTORY/FULL command and revised by the VMS BACKUP Utility. (See the "OpenVMS BACKUP Utility Reference Manual" for details.) This time is also revised by BacPac through use of the /RECORD option.

/BACKUP is only valid when used in conjunction with the /SINCE or the /BEFORE qualifier. If you use /BACKUP and /BEFORE, files with a backup date prior to the date and time specified are selected. If you use /BACKUP and /SINCE, files with a backup date equal or later than the specified date and time are selected. Files with no backup date are NOT selected. If neither /SINCE nor /BEFORE is specified, an error message is generated indicating an error in the command line.

Files which have never been marked as backed up have a system time of zero recorded in the backup date field of the file's header record. The time displayed by VMS during execution of a DIRECTORY/FULL command is <No backup done>. The negated qualifier /NOBACKUP will select these files only. If used with the /BEFORE or /SINCE qualifier, files will be selected based on the backup date as well as those files which have never been backed up.

This qualifier may not be used in conjunction with the /CREATED, /EXPIRED, or /MODIFIED qualifiers. These qualifiers are mutually exclusive.

note

Files which have the file attribute 'Backups disabled' will not be selected. This attribute is displayed by VMS during execution of a DIRECTORY/FULL command. To have these files saved, it is necessary to use the /IGNORE qualifier.

examples

```
$ BACPAC *.DOC/BACKUP/BEFORE MUA0:DOCFILES (1)
```

(1) All files in the current default directory which have file type .DOC and which have been marked as backed-up before 00:00 o'clock of the current day will be selected for processing. (See the section on /BEFORE.) The resultant pacset will be called DOCFILES.PAC.

```
$ BACPAC *.MEM/BACKUP/SINCE=12-MAY-1995 MUA0:MEM (2)
```

(2) All files in the current default directory which have file type .MEM and which have been marked as backed-up since 00:00 o'clock of May 12, 1995 will be saved. The resultant pacset will have the file name MEM.PAC.

```
$ BACPAC *.* /IGNORE=NOBACKUP MUA0:BCKFILES (3)
```

(3) All files in the current default directory, including those which have been marked as <Backups disabled>, will be saved.



/BEFORE

Requests that files be selected which are dated earlier than the date specified.

type

Input file selection qualifier.

format

/BEFORE (default is TODAY)

/BEFORE=time

qualifier value

time (See text below.)

operations

SAVE	DUPLICATE
	

related qualifiers

[/BACKUP] **[/CREATED]** **[/EXPIRED]** **[/MODIFIED]** **[/SINCE]**

restrictions

none

description

/BEFORE causes files to be selected from the input-file-spec which are dated earlier than the time specified. The file time selected from the file header is dependent upon the use of the qualifiers /CREATED, /EXPIRED, /MODIFIED, or /BACKUP. If none of these other qualifiers is present on the command line, /MODIFIED is assumed.

The time entered may be absolute time or delta time. Refer to the "OpenVMS DCL Concepts Manual" for the proper format. The time may also be selected using one of the following keywords.

BACKUP The backup file time in the file's header

TODAY The current day at 00:00:00.0 o'clock

TOMORROW 24 hours after 00:00:00.0 o'clock of the current day

YESTERDAY 24 hours before 00:00:00.0 o'clock of the current day

examples

```
$ BACPAC *.* /BEFORE MKA500:MYDIR (1)
```

(1) All files in the current default directory which have been created before 00:00 o'clock of the current day will be saved (the default for **/BEFORE** is **TODAY**). The resultant pacset will be written to the tape in MKA500. It will be called MYDIR.PAC.

```
$ BACPAC *.*;*/BEFORE=10:00 MSA0:MYDIR (2)
```

(2) All files in the current default directory which have been created before 10:00 o'clock of the current day will be saved. The resultant pacset will be written to tape in MSA0: and will be called MYDIR.PAC.

```
$ BACPAC *.* /BEFORE=BACKUP/MODIFIED MUA0:MYDIR (3)
```

(3) All files in the current default directory are candidates for inclusion in the pacset. Of the set of candidates, files will be selected which have a revision time in the file header which is earlier than the backup time for the same file.



/BLOCK_SIZE

Specifies the block size, in bytes, of the pacset to be created.

type

Save command qualifier.

format

/BLOCK_SIZE=number

qualifier value

number (default = 16384)

The number indicates the number of bytes to be contained in a single block of a pacset. Block sizes are allowed in the range 8192 to 32256 bytes.

operations



related qualifiers

none

restrictions

none

description

By default, BacPac creates pacset files containing fixed length records with a block size of 16384 bytes. This qualifier is used to override this default. Some tape drives specify optimal blocking sizes which you would want to use. Block sizes are allowed in the range 8192 to 32256 bytes. If an invalid block size is specified, a warning message is output and the save operation continues with the default record size of 16384. The block size must be a multiple of 512. If it is not, BacPac will round down to the next multiple of 512 before processing.

The last block of a file will be padded with zeros if there are not enough characters to fill it.

example

```
$ BACPAC/BLOCK_SIZE=8192 *.COM MUA0:COMFILES (1)
```

(1) All files with file type .COM in the current default directory will be saved. The resultant pacset will be called COMFILES.PAC. The pacset will be written to the tape mounted on MUA0: and will have a block size of 8192 bytes.



/BRIEF

Requests a brief listing of a pacset.

type

Command qualifier.

format

/BRIEF

operations

LIST

related qualifiers

[/BRIEF] [/EXCEPTIONS] [/LIST] [/NONE] [/SUMMARY]

restrictions

none

description

/BRIEF requests a brief listing of the contents of one or more pacsets. The listing consists of the pacset description (if input with /COMMENT during the save operation), the version of BacPac and the operating system used during the save operation, as well as the command line and date. The file specification of each file contained in the pacset as well as additional information from the file header is also provided.

example

```
$ BACPAC/LIST/BRIEF MUA0:FULLDISK (1)
```

(1) A description of the pacset will be output along with one line for each file contained therein. This lists the file name, file size and file creation date.

.....

/BROADCAST

Sends a broadcast message to all users at the start of a BacPac image mode operation.

type

Command qualifier.

format

/BROADCAST

/BROADCAST=text

/BROADCAST="text"

qualifier value

string (See text below.)

operations

SAVE	RESTORE
------	---------

related qualifiers

[/IMAGE] **[/LOCK_DEVICE]**

restrictions

none

description

/BROADCAST sends a message to all users at the start of an image mode save operation. The message indicates the type of operation, e.g., an image mode save, the disk the operation is being performed on and if the disk is being locked (/LOCK). Text provided by the initiator of the BacPac command will be appended.

If the text is to contain lowercase, blank, and/or non-alphanumeric characters, it must be enclosed in quotation marks. The maximum length of the text is 255 characters including the quotation marks if any.

example

```
$ BACPAC/LOCK/IMAGE DISK$USER: MKA500:USER_IMG-
```

```
$_ /BROADCAST="Disk back in about one hour" (1)
```

(1) A broadcast message will be sent to each user who is logged in (cluster wide) indicating that DISK\$USER is about to be locked and an image mode save performed. At the completion of the save operation, another broadcast will be sent to each user telling them the disk is now unlocked and available for use. The text in the example will be appended to the initial broadcast message.

.....

/BY_OWNER

Restores files with the owner set to the indicated UIC. This qualifier is synonymous with previous qualifier /OWNER_UIC.

type

Command qualifier.

format

```
/BY_OWNER=[uic]
```

qualifier value [uic]

The UIC may be specified as an identifier or in standard UIC format as described in the "VMS DCL Concepts Manual". The []s are required. Numbers must be input in octal, or preceded by a radix indicator (%X, %D, %O).operations

operations

RESTORE
RESTORE

related qualifiers

none

restrictions

none

description

By default, BacPac restores files with the same UIC as the UIC of the original files. If the user does not have sufficient privilege, the UIC of the current process default is used. Use of the /BY_OWNER provides a way to override the default processing and specify the UIC to be used. Again, if sufficient privilege is not provided, the process default UIC is used.

example

```
$ BACPAC DOCFILES/BY_OWNER=[USER]/PACSET *.* (1)
```

(1) The files contained in the pacset DOCFILES.PAC in the current default directory will be restored with the UIC of [USER]. All restored files will reside in the current default directory and have the same file name and type as the original files.



/CBT

Requests that restored file(s) be placed in the smallest number of sets of contiguous storage possible.

type

Restore command qualifier.

format

/CBT

operations



related qualifiers

[/CONTIGUOUS]

restrictions

none

description

/CBT (contiguous-best-try) causes the restored file(s) to be contiguous, i.e., to occupy consecutive logical blocks of storage on a best try basis. If more than three sets of contiguous blocks are required, the resulting file will not be marked as CBT.

By default, if the input file is contiguous or contiguous-best-try, BacPac will restore a file which has the same attributes. The /CBT qualifier is used to indicate contiguous attributes for other restored files.

If there is not enough contiguous space available on the output device while restoring contiguous files, an error message will be generated and the output file skipped. /NOCONTIGUOUS or /CBT may be used to override this condition, i.e., the output file created may not occupy consecutive blocks of storage.

example

\$ BACPAC DOC/CBT/PACSET *.* (1)

(1) The pacset DOC.PAC in the current default directory is to be restored. Each restored file will reside in the current default directory and will be contiguous if possible, i.e., it will occupy consecutive logical blocks of storage. If this is not possible, any remaining files will still be restored.

.....

/COMMENT

Requests that a comment be placed in the pacset for easy identification.

type

Save command qualifier.

format

/COMMENT=string

/COMMENT="string"

/COMMENT

qualifier value

string (See text below)

operations

SAVE


related qualifiers

none

restrictions

none

description

The supplied comment is kept as part of the pacset for easy identification of the contents of the pacset at a later time.

If the string is to contain lowercase, blank, and/or non-alphanumeric characters, it must be enclosed in quotation marks. The maximum length of a comment is 255 characters including the quotation marks if any.

If no string is supplied, the user will be prompted to enter one. Quotes are not necessary if using this form of the qualifier.

examples

```
$ BACPAC/COMMENT="this is a test" TEST.* TEST/PACSET (1)
```

(1) The files TEST.* in the current directory will be saved. The contents of the comment will be included in the pacset for later display using the /LIST qualifier. The quotation marks surrounding the actual comment are necessary since the comment includes blanks. The resulting pacset will be called TEST.PAC; it will reside in the current default directory.

```
$ BACPAC/COMMENT=Tuesday [000000...]*.* MKA300:TUESDAY (2)
```

(2) All files on the current disk will be saved. The resultant pacset will contain the contents of the comment (i.e., the word "TUESDAY"). Quotation marks are not necessary in this case since the comment does not contain spaces or non-alphanumeric characters, however, VMS will change the characters to upper case if not enclosed in quotation marks. The comment may be displayed using the /LIST qualifier.

The pacset will be called TUESDAY.PAC; it will reside on the tape in drive MKA300.



/COMPRESSION

Selects the compression algorithm.

type

Command qualifier.

format

```
/COMPRESSION=algorithm (default is FAST)
```

qualifier value

FAST (default)

MAXIMUM

NONE

Selects the type of compression to be used.

operations



related qualifiers



restrictions

none

description

The /COMPRESSION qualifier selects the type of compression. FAST is the default, and is suitable for most combinations of CPU and tape drive. On slower VAX systems, it may be advantageous to use NONE, which generally provides an improvement in speed over VMS BACKUP, but does not reduce the amount of tape used. NONE may also be of use in situations where the CPU is slow relative to the tape drive, and the tape drive has built-in compression. On an AXP system, or a fast VAX, MAXIMUM compression may yield the best performance because the I/O speed gained from the additional compression exceeds the speed lost due to additional processing.

example

```
$ BACPAC/COMP=MAX/IMAGE DISK$USER: DISK$BACKUP:[0,0]USER.IMG/PACSET (1)
```

(1) A disk to disk image save of the disk DISK\$USER will be performed using maximum compression. The resulting pacset will be called USER.IMG and will reside in the root directory of the disk DISK\$BACKUP.

```
$ BACPAC/COMP=NONE/IMAGE DISK$USER: MKB0:USER/REWIND/IGNORE=LABEL (2)
```

(2) A disk to tape image save of the disk DISK\$USER will be performed using no compression. The resulting pacset will be called USER.PAC on the tape mounted in MKB0:



/CONFIRM

Displays a prompt before each selected file or disk is processed.

type

Command qualifier.

format

```
/CONFIRM
```

operations



related qualifiers

none

restrictions

none

description

For non-image mode, a prompt is sent to SYS\$COMMAND before each selected file is processed. For image mode, a prompt is issued to confirm the disk and tape selection.

Information is displayed and the user must respond by typing a {Y,y} if the file/disk is to be processed. Any other response will result in the file/disk being skipped or the operation aborted.

examples

```
$ BACPAC/IMAGE/CONFIRM MKA600:/REWIND DKA0: (1)
```

(1) A prompt will be issued giving the volume name of the disk, as well as the pacset name and volume label of the tape. The user may abort the command prior to the disk being initialized, if any items are incorrect.

```
$ BACPAC/CONFIRM DOC/PACSET *.* (2)
```

(2) The files contained in the pacset DOC.PAC in the current default directory will be candidates for restoring. The following is a sample prompt which will be displayed for each candidate file:

```
SYS$SYSDEVICE:<USER>TEST.DOC;1 restore? [N]
```

If the user responds by typing a {Y,y} followed by a carriage return, the file will be restored. Any other response will result in the file being skipped.

The restored files will reside in the current default directory. The file names, types and versions will match those of the saved files.



/CONTIGUOUS

Requests that restored file(s) be placed in contiguous storage.

type

Restore command qualifier.

format**/CONTIGUOUS****/NOCONTIGUOUS****operations****related qualifiers****[/CBT]****restrictions**

none

description

/CONTIGUOUS causes the restored file(s) to be contiguous, i.e., to occupy consecutive logical blocks of storage.

By default, if the input file is contiguous, BacPac will create a file which is contiguous. The /CONTIGUOUS qualifier is used to indicate that other restored file(s) are to be contiguous.

If there is not enough contiguous space available on the output device when BacPac is creating contiguous files, an error message will be generated and the output file skipped.

/NOCONTIGUOUS is used to override this condition, i.e., the output file will be created, although it will not occupy consecutive blocks of storage.

example

```
$ BACPAC DOC/CONTIGUOUS/PACSET *.* (1)
```

(1) The pacset DOC.PAC in the current default directory is to be restored. Each restored file will reside in the current default directory and will be contiguous, i.e., it will occupy consecutive logical blocks of storage. If this is not possible, an error message will be generated, the file will be skipped, and any remaining files will be processed.

.....

/COPY

Requests that an input tape be copied to an output tape.

type

Copy command qualifier.

format

/COPY

operations



related qualifiers

[/ASSIST] [/IGNORE] [/LABEL] [/TAPE_EXPIRATION]

restrictions

none

description

/COPY is used to copy a BacPac tape. It will exactly copy one or more input tapes to one or more output tapes. The tape label and header records are copied exactly unless /LABEL is specified.

The output tape may have a different label and expiration date for archival purposes. Label processing on the output tape is the same as BacPac save operations.

The input and output tapes do not have to be the same size nor even the same type.

BACPAC/COPY takes care of this. The output tape is written such that files may be listed or restored starting at the 2nd or later tape, in the same manner as the input tape. Other means of COPYing tapes will not preserve this, e.g., VMS COPY.

Both tapes are mounted foreign.

example

```
$ BACPAC/COPY MKA300: MKA500: (1)
```

(1) The tape in device MKA300: will be copied the tape in device MKA500:. The label and other protection information will be the same. /REWIND may be used but is unnecessary since /COPY always starts at the beginning of tape.



/CREATED

Requests that files be saved based on the date the file was created.

type

Input file selection qualifier.

format

/CREATED/BEFORE=time

/CREATED/SINCE=time

operations



related qualifiers

[/BACKUP] [/BEFORE] [/EXPIRED] [/MODIFIED] [/SINCE]

restrictions

requires **/BEFORE** or **/SINCE**

description

/CREATED causes files to be selected from the input-file-spec based upon the system time recorded in the created date field of each file's header record. This time is displayed by VMS during execution of a DIRECTORY/FULL command.

This qualifier is only valid when used in conjunction with the /SINCE or the /BEFORE qualifier. If neither is specified, an error message is generated indicating an error in the command line. An error message is also generated if this qualifier is used in conjunction with the /BACKUP, /EXPIRED, or /MODIFIED qualifiers. These qualifiers are mutually exclusive.

examples

```
$ BACPAC [...] *.* /BEFORE /CREATED MKA400:MYDIR (1)
```

(1) All files in the directory tree starting at the current default directory which have been created before 00:00 o'clock of the current day will be saved to the tape in MKA400.

```
$ BACPAC *.DOC /CREATED /SINCE=YESTERDAY DOCS/PACSET (2)
```

(2) All files in the current default directory which have file type .DOC and which have been created since 00:00 o'clock of the previous day will be saved in the pacset file DOCS.PAC which will reside in the current default directory.

/DELETE

Requests that input files be deleted after successful creation of the pacset.

type

Save command qualifier.

format

/DELETE

operations



related qualifiers

[/CONFIRM] [/VERIFY]

restrictions

none

description

This qualifier requests that all input files be deleted after successful creation of the pacset. If used in conjunction with the /CONFIRM qualifier, the user is prompted for approval before each file deletion. A message is sent to SYS\$OUTPUT upon the successful deletion of each file if /LOG is also specified.

Sufficient user privilege is required to delete files.

examples

```
$ BACPAC/LOG/VERIFY *.DOC/DELETE MUA0:DOCFILES (1)
```

(1) All files in the current default directory which have file type .DOC will be saved to the tape in MUA0:. As each file is saved, a message is output to SYS\$OUTPUT (see /LOG).

After successful creation of the pacset all input files will be verified. A message is output to SYS\$OUTPUT indicating successful verification of each file (see /VERIFY). If no errors are detected, all input files will be deleted. A message is output to SYS\$OUTPUT indicating successful deletion of each file. If any errors are detected during verification, BacPac will terminate at the end of the verification pass; no input files will be deleted.



/DENSITY

Specifies the recording density of the tape to be written.

type

Save command qualifier.

format

/DENSITY=value**qualifier value**

1600	1600 bpi phase-encoded recording
6250	6250 bpi group-encoded recording
800	800 bpi non-return-to-zero inverted recording

operations**related qualifiers**

none

restrictions

none

description

Specifies the density to be used for recording to the tape. If none is specified and the tape has never been written, the system default for the requested drive will be used. Otherwise, the density will be the same as when the tape was last written.

If the tape is mounted before execution of BacPac, the density from the MOUNT command will be used.

examples

```
$ BACPAC/DENSITY=6250/IMAGE DISK$TEST: MKA500:TEST (1)
```

(1) An image save of the disk DISK\$TEST will be performed. The pacset will be written to the tape on drive MKA500 at a recording density of 6250 bpi.

**/DIFFERENTIAL**

Requests that BacPac perform a differential save operation.

type

Command qualifier.

format

/DIFFERENTIAL

operations



related qualifiers

[/IMAGE] [/IGNORE] [/INCREMENTAL] [/RECORD] [/SINCE]

restrictions

none

description

A differential save operation using **/DIFFERENTIAL** is analogous to an image mode save operation. The entire disk is scanned. In image mode, all files are saved. In differential mode, files are saved if they meet the additional criteria of **/SINCE=BACKUP**.

/DIFFERENTIAL is equivalent to specifying the entire disk as a file spec (**[000000...]*.*;***) and specifying **/SINCE=BACKUP**. Using the qualifier **/DIFFERENTIAL** is much faster.

example

```
$ BACPAC/DIFFERENTIAL DISK$USER1:MKA600:USER1_INC/REWIND (1)
```

(1) All files on the disk **DISK\$USER1** will be scanned. Those files whose modified date is later than the backup date will be saved. Any files whose backup date has not yet been set will also be saved.



/DISMOUNT

Specifies the output disk be dismounted before initiating an image mode duplicate or remote operation. The disk to be initialized and restored must be mounted foreign. **/DISMOUNT** provides for first dismounting a mounted disk. **/MOUNT_FOREIGN** is also required to mount the disk foreign.

type

Command qualifier.

format

/DISMOUNT

operations



related qualifiers

`[/DUPLICATE]` `[/IMAGE]` `[/MOUNT_FOREIGN]`

restrictions

none

description

On an image mode restore or image mode duplication, the disk to be initialized and restored must be mounted foreign. This qualifier is provided to facilitate remote disks. `/DISMOUNT` provides for first dismounting a mounted disk. `/MOUNT_FOREIGN` is also required to mount the disk foreign before the image mode operation may be initiated.

Both qualifiers (`/DISMOUNT` and `/MOUNT_FOREIGN`) are required if the disk to be restored is mounted for use.

examples

```
$ BACPAC/DUPLICATE/DISMOUNT/MOUNT_FOREIGN/IMAGE DISK$USER1: node::DKA0: (1)
```

(1) The disk in drive DKA0: on node node:: will be dismounted clusterwide. The disk will then be mounted foreign before it is initialized. All files from DISK\$USER1: will be duplicated on DKA0: on node node::.

```
$ BACPAC/IMAGE/DISMOUNT/MOUNT_FOREIGN DKA0: node::userdisk/PAC (2)
```

(2) An image mode restore will be performed on the local disk DKA0: using the pacset userdisk.pac on node node::. The local disk DKA0: will first be dismounted clusterwide and then mounted foreign before the restore actually takes place.



/DUPLICATE

Invokes the duplicate operation.

type

Duplicate command qualifier.

format

/DUPLICATE

operations



related qualifiers

none

restrictions

none

description

/DUPLICATE is used to invoke the duplicate operation of BacPac. BacPac supports duplicate operations for image mode as well as file mode. Image mode copies the input disk and restores and initializes the output disk. File mode copies whatever files are given in the input-file-spec to the output-file-spec.

example

```
$BACPAC/DUPLICATE DKA300:[user]*.DAT node::DKA200:[user] (1)
```

(1) The files with file type .DAT in the [user] directory on the local node will be duplicated in the [user] directory on the remote node node::.



/ENCRYPT

Requests AES encryption for the operation..

type

Command qualifier.

format

/ENCRYPT

/ENCRYPT=(name=keyname,algorithm=algo)

/ENCRYPT=(name=keyvalue,algorithm=algo)

qualifier value

keyname

name of a key defined used the VMS ENCRYPT command

keyvalue

actual key value to be used during encryption

algo

AESCFB128 (default)
 AESCFB192
 AESCFB256
 AESOFB128
 AESOFB192
 AESOFB256

operations



related qualifiers

none

restrictions

requires VMS 8.3 or greater

description

The /ENCRYPT qualifier specifies the encryption key to be used and optionally specifies a particular encryption algorithm to be used. type of compression. The encryption key and the algorithm used must be the same for a restore operation as was used for the original save operation. If no key is specified, you will be prompted to enter a key.

Keys may be created in advance of using BacPac by using the VMS ENCRYPT command. If the key value is specified, it must be enclosed in quotation marks.

BacPac supports 6 different AES algorithms. The algorithms supported are cipher feedback and output feedback for 3 different key lengths. These algorithms and the key lengths required are described under the VMS ENCRYPT command.

BacPac encrypts all data to be saved, including all file header information.

It is the responsibility of the user to remember the key. The key used is NOT stored in the pacset. It is required to either list or restore the pacset. BacPac encrypts a small test and remembers the checksum of the encrypted text. It also does this on a restore operation to verify that the key supplied is the correct key.

example

```
$ BACPAC/ENCRYPT=NAME=mykey/IMAGE DISK$USER:
DISK$BACKUP: [0,0] USER.IMG/PACSET (1)
```

(1) A disk to disk image save of the disk DISK\$USER will be performed using encryption. The key used is called mykey and has been previously defined using the VMS ENCRYPT command.

The resulting pacset will be called USER.IMG and will reside in the root directory of the disk DISK\$BACKUP.

```
$ BACPAC/ENCRYPT= (NAME=mykey, ALGORITHM=AESCFB256) /IMAGE DISK$USER:
MKB0:USER (2)
```

(2) A disk to tape image save of the disk DISK\$USER will be performed using the AESCFB256 encryption algorithm. The key used will be mykey. It must have been defined with a length long enough to satisfy the algorithm. The resulting pacset will be called USER.PAC:.



/EXCEPTIONS

Requests display of exception status in a pacset.

type

Command qualifier.

format

/EXCEPTIONS (default)

/NOEXCEPTIONS

operations

LIST


related qualifiers

[/BRIEF] **[/FULL]** **[/LIST]** **[/NONE]** **[/SUMMARY]**

restrictions

none

description

/EXCEPTIONS displays messages regarding files which were open during the save operation and files which are marked 'Backups Disabled', i.e., files which may not be restored in full.

example

```
$ BACPAC/LIST/EXCEPTIONS/NONE MUA0:MYDIR (1)
```

Status messages only will be displayed (/NONE).



/EXCLUDE

Requests that the specified files be excluded during file selection.

type

File selection qualifier.

format

/EXCLUDE=exclude-file-spec

/EXCLUDE=(exclude-file-spec,...)

qualifier value

file specification list of the files to be excluded. There is no default value.

operations

SAVE	RESTORE	LIST	DUPLICATE
█	█	█	█

related qualifiers

[/SELECT]

restrictions

none

description

During a save operation, this qualifier is used to exclude files in the input-file-spec from processing. It is particularly useful when the input-file-spec contains wild-cards. There is no default for the file name or file type. Relative version numbers are not supported.

The exclude-file-spec may contain a list of file specifications. In this case, the list must be delimited by parentheses. Wild cards and sticky defaults are supported.

During a restore operation, this qualifier is used to exclude files in the pacset from being restored.

examples

```
$ BACPAC *.*;*/EXCLUDE=(*.COM,*.DAT)MYFILES/PACSET (1)
```

(1) All files in the current default directory will be selected for saving with the exception of any file which has file type .COM or file type .DAT. The created pacset will be called MYFILES.PAC and will reside in the current default directory.

.....

/EXPIRED

Requests that files be selected based on the expiration date of the file.

type

Input file selection qualifier.

format

/EXPIRED/BEFORE=time

/EXPIRED/SINCE=time

/NOEXPIRED

operations

SAVE	DUPLICATE
<input type="checkbox"/>	<input type="checkbox"/>

related qualifiers

[/BACKUP] [/BEFORE] [/CREATED] [/MODIFIED] [/SINCE]

restrictions

requires **/BEFORE** or **/SINCE**

description

/EXPIRED causes files to be selected from the input-file-spec based upon the system time recorded in the expiration date field of each file's header record. This time is displayed by VMS during execution of a DIRECTORY/FULL command. This field may be set explicitly using a SET FILE command or implicitly using a SET VOLUME command (See the "OpenVMS DCL Dictionary" for details).

This qualifier is only valid when used in conjunction with the /SINCE or the /BEFORE qualifier. If neither is specified, an error message is generated indicating an error in the command line. An error message is also generated if this qualifier is used in conjunction with the /BACKUP, /CREATED, or /MODIFIED qualifiers. These qualifiers are mutually exclusive.

Files which have not been set to expire have a system time of zero recorded in the expired date field of the file's header record. The time displayed by VMS during execution of a DIRECTORY/FULL command is <None specified>. Use of the /EXPIRED qualifier in conjunction with /BEFORE or /SINCE will result in these files never being selected. The negated qualifier /NOEXPIRED will select these files only.

This negated qualifier does not require the use of the /SINCE or /BEFORE qualifier. If one is specified, it will be ignored. An error message will however be generated if this qualifier is used in conjunction with the /BACKUP, /CREATED, or /MODIFIED qualifiers. These qualifiers are mutually exclusive.

examples

```
$ BACPAC *.* /BEFORE/EXPIRED MYDIR/PACSET (1)
```

(1) All files in the current default directory which have an expiration time before 00:00 o'clock of the current day will be selected. The resultant pacset will be called MYDIR.PAC which will reside in the current default directory.

```
$ BACPAC *.*;*/SINCE=TODAY/EXPIRED MYDIR/PACSET (2)
```

(2) All files in the current default directory which have an expiration time later than 00:00 o'clock of the current day will be selected. The pacset will be call MYDIR.PAC; it will reside in the current default directory.

```
$ BACPAC *.*;*/NOEXPIRED MYDIR/PACSET (3)
```

(3) All files in the current default directory which have no expiration time will be included in the pacset called MYDIR.PAC which will reside in the current default directory.

.....

/FULL

Requests a detailed listing of a pacset.

type

Command qualifier.

format

/FULL

operations

LIST
████████

related qualifiers

[/BRIEF] [/COMMENT] [/EXCEPTIONS] [/LIST] [/NONE] [/SUMMARY]

restrictions

none

description

/FULL requests a complete listing of the contents of one or more pacsets. The listing consists of the pacset description (if input with /COMMENT during the save operation), the version of BacPac and the operating system used during the save operation as well as the command line and date.

The file specification of each file contained in the pacset as well as additional information from the file header is also provided.

example

```
$ BACPAC/LIST/FULL DKA0:[BACKUPS] FULLDISK (1)
```

(1) The header of the pacset FULLDISK.PAC will be displayed to SYS\$OUTPUT:

.....

/IGNORE

Requests that BacPac ignore certain file processing restrictions.

type

Command qualifier.

format

/IGNORE=option

qualifier value

option (See text below.)

operations



related qualifiers

[/BACKUP] [/DIFFERENTIAL] [/IMAGE] [/INCREMENTAL] [/LABEL]

restrictions

none

description

BacPac processing restrictions may be overridden by using the following options:

ACCESSIBILITY

By default, BacPac will not write to a tape whose accessibility field contains a character other than blank. Usually this is set by a system other than VMS.

INTERLOCK

By default, BacPac will not save files which are currently open by another process or user. This option can be used to override this restriction and process files that are currently open for reading or writing. Use of this option requires the privilege SYSPRV, a system UIC, or ownership of the volume.

This is defaulted on for /DIFFERENTIAL, /INCREMENTAL, and /IMAGE.

LABEL_PROCESSING

By default, BacPac checks the tape label, tape expiration date, accessibility field, and protection fields of any tape that is to be written. This option can be used to override these restrictions except for the tape accessibility field.

NOBACKUP

By default, BacPac will not process files which have the file attribute 'Backups disabled'. This attribute is displayed by VMS during execution of a DIRECTORY/FULL command. To have these files processed, it is necessary to use the /IGNORE qualifier. (See also /BACKUP.)

example

```
$ BACPAC [*...]/IGNORE=NOBACKUP MKA500:ALLDIRS (1)
```

(1) All files on the current disk will be selected. Files which have the file attribute 'Backups disabled' will be processed as though the attribute was not set. The pacset will be called ALLDIRS.PAC and will be written to the tape in device MKA500.

```
$ BACPAC/IMAGE/IGNORE=INTERLOCK DISK$USER1: MKA300:USER1_IMG (2)
```

(2) An image backup of the disk DISK\$USER1 will be done. All files which are open by another user will be processed as though the file were not open. The pacset will be called USER1_IMG.PAC and will be written to the tape mounted in MKA300:.

```
$ BACPAC/IMAGE/IGNORE=( INTERLOCK, LABEL, ACCESS) -
```

```
$ _ DISK$USER1: MKA300:USER1_IMG/LABEL=USER (3)
```

(3) The same image backup as in (2) is done. A new label is written to the tape and additional processing is ignored, i.e., label and protection checking including the accessibility field.

```
$ BACPAC/IMAGE/IGNORE=( INTERLOCK, LABEL, ACCESS) -
```

```
$ _ DISK$USER1: MKA300:USER1_IMG/LABEL=USER (3)
```

(3) The same image backup as in (2) is done. A new label is written to the tape and additional processing is ignored, i.e., label and protection checking including the accessibility field.

```
$ BACPAC/COPY/IGNORE=LABEL MKA300: MKA500:/LABEL=USER (4)
```

(4) The tape in device MKA300: will be copied to the tape in device MKA500. The output tape will be labeled USER and no OPCOMs or prompts will be generated.

.....

/IMAGE

Requests that BacPac perform an image mode save or restore operation.

type

Command qualifier.

format

/IMAGE

/IMAGE=SAVE

/IMAGE=RESTORE

qualifier value

SAVE indicates this is a SAVE operation

RESTORE indicates this is a RESTORE operation

operations



related qualifiers

[/DIFFERENTIAL] **[/IGNORE]** **[/INCREMENTAL]**

restrictions

none

description

An image mode operation saves or restores an entire disk. On a save operation, volume information is written to the pacset. The qualifier **/IGNORE=INTERLOCK** is set by default for image save operations. On a restore operation, this information is used to initialize the disk before restoring files. File specifications are not allowed.

If the qualifier value is specified, BacPac will verify the devices for consistency with the specified value. An error message will be generated if the qualifier value and the device types are inconsistent, e.g., **/IMAGE=SAVE** has been specified and the input device is a tape device.

example

```
$ BACPAC/DUPLICATE/IMAGE DISK$USER1: node::DKA100: (1)
```

(1) All files on the disk DISK\$USER1 will be duplicated to the disk DKA100: on node node::, including those which are open by another user. The target disk needs to be mounted foreign before invoking this operation. The target disk will be initialized using information from the source disk.

```
$ BACPAC/IMAGE/LOG/CONFIRM MKA300:USER1_IMG DISK$USER1: (2)
```

(2) An image restore of the disk DISK\$USER1 will be done. Prior to initialization of the disk, the user will be prompted to confirm the disk selection. The prompt contains the volume name of the disk to be restored, the name of the pacset and the label of the tape from which the disk will be restored. The disk will then be initialized using the information contained in the pacset USER1_IMG.PAC. All files will then be restored. The disk will be virtually the same as before the save operation took place.

.....

/INCREMENTAL

Requests that BacPac perform an incremental save or restore operation.

type

Command qualifier.

format

/INCREMENTAL

operations

SAVE	RESTORE	DUPLICATE
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

related qualifiers

[/DIFFERENTIAL] [/IGNORE] [/IMAGE] [/RECORD] [/SINCE]

restrictions

none

description

An incremental save operation using /INCREMENTAL is analogous to an image mode save operation. The entire disk is scanned. In image mode, all files are saved. In incremental mode, files are saved if they meet the implied selection criteria /SINCE=BACKUP. Processing of alias files is the same as with image mode. /RECORD and /IGNORE=INTERLOCK are also implied qualifiers.

To perform an incremental restore operation, an image mode restore operation must have just been performed on a disk. This qualifier provides for restoring the incremental PacSets generated after the image mode pacset. Files which have been deleted since the image mode

save operation, will be deleted. New files will be restored. Changed files will be updated. This is part of the process of restoring a disk to its state prior to the last incremental save operation. File specifications are not allowed. (Also see the section 'Incremental Backups with BacPac'.)

example

```
$ BACPAC/SINCE=BACKUP/INCREMENTAL DISK$USER1: -
```

```
$ _ MKA600:USER1_INC/REWIND/RECORD (1)
```

(1) All files on the disk DISK\$USER1 will be scanned. Those files whose modified date is earlier than the backup date will be saved. Any files whose backup date has not yet been set will also be saved. All saved files will have their backup date changed to that of the time of the BacPac operation. The record pass is much faster using /INCREMENTAL than with a similar command.

```
$ BACPAC/INCREMENTAL MKA600:USER1_INC/REWIND DISK$USER1: (2)
```

(2) The files contained in the incremental pacset will be restored. Files which have been deleted prior to the generation of the incremental pacset, will be deleted from the disk. Once all incremental PacSets have been applied, the disk will look virtually the same as before the last incremental pacset was generated.



/INITIALIZE

Requests that BacPac initialize the disk using parameters from the image mode PacSet during an image mode restore operation.

type

Command qualifier.

format

/INITIALIZE (default)

/NOINITIALIZE

operations

RESTORE	DUPLICATE
---------	-----------

related qualifiers

[/IMAGE]

restrictions

none

description

During an image mode restore operation, the output disk is always initialized. This qualifier controls whether the initialization parameters come from the pacset or from the disk. The default is to take the parameters from the pacset. This may not be advisable if the output disk is of a different type or size. In this case, the user would initialize the disk using the VMS INITIALIZE command and then restore the pacset using the negated form of the qualifier.

example

```
$ BACPAC/IMAGE MKA600:USER1_IMG/REWIND DISK$USER1:/INITIALIZE (1)
```

(1) The disk DISK\$USER1 will be restored using an image mode restore operation. The disk will be initialized using the parameters from the pacset USER1_IMG.PAC. After initialization, all files will be restored to the disk. The disk will be virtually a copy of what it was before the image mode save operation.

```
$ BACPAC/IMAGE/LOG MKA300:USER1_IMG DISK$USER1:/NOINITIALIZE (2)
```

(2) An image restore of the disk DISK\$USER1 will be done. The disk will first be initialized using the information contained on the disk itself. The disk should have been initialized prior to this restore operation. Parameters are checked against those in the pacset to verify all restored files will in fact fit on the disk. All files will then be restored. The number and content of the files on the disk will be virtually the same as before the save operation took place.

**/JOURNAL**

Controls the writing or reading of a journal file which contains a list of the contents of one or more BacPac save operations.

type

Command qualifier.

format

/JOURNAL

/JOURNAL=journal-file-spec

qualifier value

file specification for the journal file. Default is BACPAC.PJL.

operations

SAVE	LIST
	

related qualifiers

[/LIST]

restrictions

none

description

During a save operation, the pacset name as well as the file name of each file saved is written to a journal file. If a journal-file-spec is specified and does not exist, one is created. If the journal file does exist, it is opened and appended to. The journal file may reside on any node in the network. If the journal file is located on another node, the node name must be part of the file specification.

The journal file is a running list of the contents of several PacSets. It might be used, for example, as a weekly journal of the contents of incremental PacSets. This can provide a quick way to find a file which needs to be restored.

BacPac can also be used to read journal files created by VMS BACKUP.

The journal-file-spec follows standard VMS conventions for file specifications. The default device and directory will be the current process defaults; the default file name and type will be BACPAC.PJL.

The contents of a journal file may be read using the /LIST qualifier.

examples

```
$ BACPAC/JOURNAL [...] *.* MUA0:DIRTREE/REWIND (1)
```

(1) The directory tree starting at the current default directory will be written to the PacSet DIRTREE.PAC on the tape mounted in device MUA0: The pacset name as well as the names of all files contained in the PacSet will be written to the journal file BACPAC.PJL in the current default directory.

```
$ BACPAC/LIST/JOURNAL (2)
```

(2) The contents of the journal file BACPAC.PJL in the current default directory will be displayed to SYS\$OUTPUT.

.....

/KEEP

Specifies the name of the pacset to be used in the duplicate operation.

type

Command qualifier.

format

/KEEP=pacset_name**operations****related qualifiers**

[/DUPLICATE]

restrictions

none

description

A BacPac duplicate operation may optionally write a pacset. The pacset must reside on a disk available to the remote system. It cannot be the same disk that the duplication is occurring on.

examples

```
$ BACPAC DISK$USER1: BRAVO::$3$DKA0: /DUP /IMAGE /MOUNT
/KEEP=DKA100:user1
```

(1) The local disk DISK\$USER1 will be duplicated onto \$3\$dka0 on node BRAVO. A pacset will be written in addition to restoring the local disk on the remote system. The pacset will be called user1.pac and will reside on disk DKA100 on node BRAVO..

**/LABEL**

Specifies the tape label to be used during tape label processing.

type

Command qualifier.

format

/LABEL=tape-label[,tape_label,...]

qualifier value

6 character string for the tape-label. Default is BACPAC.

operations

related qualifiers

`[/IGNORE]` `[/REWIND]`

restrictions

none

description

Specifies the label(s) to be written to the tape or verified against what is already written on the tape. The label cannot exceed 6 characters. If no label is supplied, BacPac uses the label BACPAC. This default may also be overridden using the logical name **BACPAC_LABEL**. During a **COPY** operations, the default is the label from the input tape.

The tape label is written to the first record on a tape. It is distinguished from the pacset name which is the name of a file on the tape. There may be many pacsets per tape.

If a pacset spans several physical tapes, the tape label is written to the first tape as supplied by the user. The label written to the second and subsequent tapes is created from the first 4 characters of the label and 2 digits starting with the characters 02. If the user has supplied a list of labels, these will be used in the order supplied. In this case, the two characters indicating the volume number are not replaced in the label. If the number of tapes exceeds the labels provided, the first label is used with the volume number appended for subsequent tapes.

examples

```
$ BACPAC/REWIND/LABEL=MYTAPE [...] *.* MUA0:DIRTREE (1)
```

(1) All the files in the directory tree starting at the current default directory will be saved in the pacset DIRTREE.PAC on the tape which resides in device MUA0:. The tape is to be initialized (/REWIND). The label which will be written to the tape is MYTAPE. All label checking will first be done before the tape is initialized. This of course is bypassed if the tape is new.

```
$ BACPAC/LABEL=MYTAPE [OTHERDIR...] *.* MUA0:OTHERDIR (2)
```

(2) This is the same physical tape as in Example 1. Another pacset will be added to it. The tape label is verified before writing the new pacset.

```
$ BACPAC/REWIND/LABEL=(USER1,USER2) [USERDISK...] *.* -
```

```
$ _ MUA0:USER_INC (3)
```

(3) This example creates a pacset of the disk USERDISK. It will be written to the tape residing in MUA0. The first tape will be labeled USER1. If a second tape is required, it will be labeled USER2. If a third tape is required, it will be labeled USER03.

```
$ BACPAC/COPY/LABEL=ARCHV MKA300: MUA0: (4)
```

(4) This example copies the tape in device MKA300: to the tape in device MUA0:. The newly copied tape will have the label ARCHV no matter what the label of the input tape. The two tapes will be virtually identical in all other respects.



/LIST

Display the contents of a pacset to SYS\$OUTPUT or a file.

type

Command qualifier.

format

/LIST

/LIST=list-file-spec

qualifier value

file specification for the listing file. Default file type is .LIS.

operations



related qualifiers

[/BRIEF] [/EXCEPTIONS] [/FULL] [/LOG] [/NONE] [/VERIFY]

restrictions

none

description

Displays the contents of a pacset; by default, the display is written to SYS\$OUTPUT. If a list-file-spec is provided, the display is written to the file. The list-file-spec follows standard VMS conventions for file specifications. A node name may be specified. The default device and directory will be the current process defaults; the default type will be .LIS. There is no default for the file name; the user must supply one.

examples

```
$ BACPAC/LIST MUA0:DIRTREE (1)
```

(1) The contents of the pacset DIRTREE.PAC will be displayed.

```
$ BACPAC/LIST=DIRTREE MUA0:DIRTREE (2)
```

(2) This is the same as Example 1 except that a file is created. DIRTREE.LIS will be reside in the current default directory.

```
$ BACPAC/LIST=DIRTREE/LOG [...]*. * MUA0:DIRTREE (3)
```

(3) This is similar to Example 1 except that the listing file is created at the same time as the pacset. It is called DIRTREE.LIS and will be reside in the current default directory. The listing is not displayed as the command executes, however, since /LOG is also specified, messages are displayed as each file is saved.



/LOCK_DEVICE

Software write locks a disk during an image mode save.

type

Save command qualifier.

format

/LOCK_DEVICE

operations



related qualifiers

[/BROADCAST][/IMAGE]

restrictions

should not be used on a system disk

description

/LOCK_DEVICE requests that the disk being saved be locked against write access by users for the duration of the save. This effectively offers the same protection against change as running standalone but without the necessity to shutdown the system. Users still have read access to the disk. At the completion of the save operation, before the record pass if any, the disk is unlocked. A CTRL-C or CTRL-Y will abort any save in progress and unlock the disk.

This qualifier should not be used on a system disk. The system may hang, requiring a reboot. You may not even be able to type a CTRL-Y.

examples

```
$ BACPAC/LOCK_DEVICE/IMAGE=SAVE/BROADCAST DUA1: MUA0:DUA1_IMG (1)
```

(1) The user disk DUA1: will be locked against write access during the image mode save to the tape in MUA0. All users will be notified at the time the image save commences (/BROADCAST).



/LOG

Display progress and status messages to SYS\$OUTPUT.

type

Command qualifier.

format

/LOG

/LOG=log-file-spec

operations

SAVE	RESTORE	LIST	DUPLICATE
------	---------	------	-----------

related qualifiers

[/STATISTICS]

restrictions

none

description

Displays the file specification of each input file processed as the command executes; the display is written to SYS\$OUTPUT.

/LOG implicitly implies /STATISTICS. If statistics are not desired, this must be explicitly requested using the negated qualifier /NOSTATISTICS.

examples

```
$ BACPAC/LOG [...] *.* MUA0:DIRTREE (1)
```

(1) The directory tree starting with the current default directory will be saved in the PacSet DIRTREE.PAC on the tape residing in MUA0:. The following is a sample of the information which is sent to SYS\$OUTPUT during execution of this command:

```
%BACPAC-S-SAVED, saved SYS$SYSROOT:[SYSEXE] SYSUAF.DAT;2
```



/MEDIA_FORMAT

Turns on hardware data compaction for the tape device.

type

Save command qualifier.

format

/MEDIA_FORMAT=[NO]COMPACTION

operations



related qualifiers

[/COMPRESSION]

restrictions

none

description

/MEDIA_FORMAT controls whether data records are compacted by the tape device. Using this qualifier with compression on for BacPac will not make the operation run faster, will not decrease the amount of tape required and may actually increase the time. This qualifier is best used with compression turned off for BacPac. Results will then be similar to those obtained using VMS Backup.

examples

\$ BACPAC/MEDIA=COMPACT/COMPRESSION=NONE/IMAGE DUA1: MUA0:DUA1_IMG (1)

(1) The user disk DUA1: will be saved without compression by BacPac but hardware compaction for the tape device MUA0 will be turned on.



/MODIFIED

Requests that files be selected based on the date of last file modification.

type

Input file selection qualifier.

format

/MODIFIED/BEFORE=time

/MODIFIED/SINCE=time

operations



related qualifiers

[/BACKUP] [/BEFORE] [/CREATED] [/EXPIRED] [/SINCE]

restrictions

requires **/BEFORE** or **/SINCE**

description

/MODIFIED causes files to be selected from the input-file-spec based upon the system time recorded in the revised date field of each file's header record. This time is displayed by VMS during execution of a DIRECTORY/FULL command.

This qualifier is only valid when used in conjunction with the /SINCE or the /BEFORE qualifier. If neither is specified, an error message is generated indicating there is an error in the command line. The same is true if this qualifier is used in conjunction with the /BACKUP, /CREATED, or /EXPIRED qualifiers. These qualifiers are mutually exclusive.

examples

```
$ BACPAC [...] *.*;0/BEFORE/MODIFIED MYDIR/PACSET (1)
```

(1) All files in the directory tree beginning with the current default directory (highest version only) which have a revision time in the file header before 00:00 o'clock of the current day will be saved. The resultant PacSet will be called MYDIR.PAC; it will reside in the current default directory.

```
$ BACPAC *.* /SINCE=(5-JAN-1987:9:00) /MODIFIED/LOG MUA0:DIRSAVE (2)
```

(2) All files in the current default directory which have a revision time later than 09:00 o'clock of January 5, 1987 will be saved. The pacset will be called DIRSAVE.PAC and will be written to the tape in MUA0:.

/MOUNT_FOREIGN

Specifies the target disk of an image mode duplication be mounted foreign prior to the operation commencing. The disk to be initialized and restored must be mounted foreign.

type

Command qualifier.

format

/MOUNT_FOREIGN

operations



related qualifiers

[/DISMOUNT] [/IMAGE] [/REMOUNT]

restrictions

none

description

On an image mode restore or image mode duplication, the disk to be initialized and restored must be mounted foreign. This qualifier is provided to facilitate remote disks. /DISMOUNT provides for first dismounting a mounted disk. /MOUNT_FOREIGN is also required to mount the disk foreign before the image mode operation may be initiated.

Both qualifiers (/DISMOUNT and /MOUNT_FOREIGN) are required if the disk to be restored is mounted for use.

examples

```
$ BACPAC/DUPLICATE/DISMOUNT/MOUNT_FOREIGN/IMAGE DISK$USER1: node::DKA0: (1)
```

(1) The disk in drive DKA0: on node node:: will be dismounted clusterwide. The disk will then be mounted foreign before it is initialized. All files from DISK\$USER1: will be duplicated on DKA0: on node node::.

```
$ BACPAC/IMAGE/DISMOUNT/MOUNT_FOREIGN DKA0: node::userdisk/PAC (2)
```

(2) An image mode restore will be performed on the local disk DKA0: using the pacset userdisk.pac on node node::. The local disk DKA0: will first be dismounted clusterwide and then mounted foreign before the restore actually takes place.



/NAME

Requests that BacPac use the specified logical name when mounting the disk after an image mode replication operation.

type

Command qualifier.

format

/NAME=logical-name

qualifier value

logical name to be used for the newly restored disk

operations

DUPLICATE

related qualifiers

[/REMOUNT]

restrictions

none

description

After an image mode duplication, if /REMOUNT is specified, BacPac will mount the newly restored disk. The specified logical name will be used during the mount operation.

examples

```
$ BACPAC/DUPLICATE/IMAGE/REMOUNT/NAME=DISK$USER1 DKA0: node::DKA100: (1)
```

(1) An image mode duplication will be done to the disk DKA100: on node node::. At completion of the operation, BacPac will mount the disk DKA100: for cluster and system use and will establish the logical name DISK\$USER1 for this disk.

/NEW_VERSION

Requests that restored files with matching file names be given new (higher) version numbers.

type

Restore command qualifier.

format

/NEW_VERSION

/NONEW_VERSION (default)

operations



related qualifiers



restrictions

none

description

/NEW_VERSION allows identically named files to be restored, setting the version number of the restored file to the highest existing number plus one. By default, BacPac will not restore a file with an identical file name to one that already exists; an error message is generated.

Use of the /NEW_VERSION qualifier is one way to override this default. /REPLACE and /OVERLAY are others.

By default, restored files have the same version number as the original file. If an identically named file exists with a higher version number, the saved file is restored (with the lower version number) and a warning message is generated. The use of /NEW_VERSION will give the newly restored file a version number higher than any existing file of the same name and type rather than lower. In this case, no warning message is generated. The table below gives an example of this use of /NEW_VERSION.

Saved File	Existing File	Default Result	Result using /NEW_VERSION
file.exp;3	file.exp;4	file.exp;3 (warning message)	file.exp;5
file.exp;3	file.exp;3	no file expanded (error message)	file.exp;4
file.exp;3	file.exp;2	file.exp;3	file.exp;3

note

If multiple versions of a file exist in the pacset using /NEW_VERSION will cause the version numbers to be reversed. For example, a file with version numbers ;5, ;4, ;3 and ;2 and which does not currently exist on disk, will get restored with version numbers ;5, ;6, ;7, and ;8.

example

```
$ BACPAC MYFILES/PACSET/NEW_VERSION *.* (1)
```

(1) The files contained in the pacset MYFILES.PAC in the current default directory will be restored. If files with the same version number exist for the files contained in MYFILES.PAC, the saved files will be restored and given a higher version number. All restored files will reside in the current default directory and have the same file name and type as the corresponding saved files.



/NONE

During a LIST operation, requests that no list output be generated. Only errors or status messages are reported.

type

Command qualifier.

format

/NONE

operations

LIST
■

related qualifiers

[/EXCEPTIONS] [/LIST] [/LOG] [/VERIFY]

restrictions

none

description

During a **LIST** operation (/LIST), if /VERIFY is also selected, each file in the pacset is expanded in memory with CRC checking. By default, a listing is produced. /NONE is used to verify the contents of a pacset only. Only error messages and exceptions (/EXCEPTIONS) will be produced.

During a **SAVE** or **RESTORE** operation, /NONE is used to get a listing of exceptions only, i.e., those files which were open during the save operation or which are marked 'Backups Disabled'.

example

\$ BACPAC/LIST/NONE/VERIFY MYDIR/SAVE (1)

(1) All files in the pacset MYDIR.PAC, residing in the current default directory, will be expanded in memory with CRC checking. No list output will be produced.



/NOTIFY

Sends a VMS MAIL notification to a specified user at the completion of a BacPac operation.

type

Command qualifier.

format

/NOTIFY

/NOTIFY=username

qualifier value

VMS username of the intended recipient. Default is the BacPac initiator.

operations

SAVE	RESTORE	LIST
------	---------	------

related qualifiers

none

restrictions

none

description

At the completion of a BacPac operation, a VMS MAIL message is sent to a specified user, or if none is specified, to the user who initiated the BacPac command. The message will contain the BacPac error log if one was generated. If there were no errors, the message will contain the command line used and a success notification.

example

\$ BACPAC/IMAGE/NOTIFY DISK\$USER: MKA300:USER_IMG (1)

(1) An image save will performed on the disk DISK\$USER. A MAIL notification will be sent to the initiator of the BacPac when the save is finished.

/OVERLAY

Requests that restored files with matching file names be written over the existing files.

type

Restore command qualifier.

format

/OVERLAY

operations



related qualifiers

[/NEW_VERSION] [/REPLACE] [/TRUNCATE]

restrictions

none

description

/OVERLAY overwrites existing files with identically named files from the pacset. By default, BacPac will not restore a file with an identical file name to one that already exists; an error message is generated.

Use of the /OVERLAY qualifier is one way to override this default. /REPLACE and /NEW_VERSION are others. /OVERLAY is similar to /REPLACE but with different placement of the file on disk.

/OVERLAY causes the blocks associated with a file to be overwritten with the contents of the file being restored. If the restored file contains more blocks than the file it is overlaying, the file will be extended as necessary. If the restored file contains fewer blocks than the file it is overlaying, the end of file marker is reset, unless /TRUNCATE is also specified in which case the file is truncated.

example

```
$ BACPAC MYFILES/PACSET/OVERLAY/SELECT=LARGE.DAT *.* (1)
```

(1) The file LARGE.DAT contained in the pacset MYFILES.PAC in the current default directory will be restored. If a file with the same version number exists, the saved file will be written on top of the existing file.

/PACSET

Specifies the name of the pacset to be used in the operation. This is required for disk to disk operations.

type

Command qualifier.

format

/PACSET

operations



related qualifiers

none

restrictions

none

description

A BacPac save operation writes a pacset a BacPac restore operation reads a pacset. It is imperative in disk-to-disk operations that the user specify which file specification is the pacset and thus whether the operation is a save operation or a restore operation. This qualifier is optional when a tape is involved. It is also not necessary when listing a pacset.

examples

```
$ BACPAC [...] *.* DIRTREE/PACSET (1)
```

(1) The directory tree starting at the current default directory will be saved to the pacset DIRTREE.PAC which also resides in the current default directory.



/PASSWORD

Specifies a password for the resultant pacset.

type

Command qualifier.

format

/PASSWORD**operations****related qualifiers**

none

restrictions

none

description

This qualifier is used to add password protection to the pacset. The user will be prompted (SYSS\$COMMAND) to enter the password once execution has begun. The password is NOT displayed as the user types it. A verification prompt is also issued. The password may be any character string up to 80 characters in length. The password will subsequently be required by the restore operation.

The user assumes full responsibility for remembering the password. **It cannot be read directly from the pacset. Once a password is used in creation of a pacset, the password MUST be supplied for later operations with the pacset.**

example

```
$ BACPAC/IMAGE/PASSWORD DISK$USER: MKA500:USER (1)
```

(1) An image mode pacset will be created for the disk DISK\$USER. A prompt will be issued for a password. **The password is NOT displayed as the user types it.** The password is stored as part of the image pacset USER.PAC. The password MUST be entered for any restore operations to proceed.

**/PRIORITY**

Requests the BacPac process to execute at the specified priority.

type

Command qualifier.

format

/PRIORITY=number

qualifier value

number

The number indicates the priority level at which the BacPac process should execute.

operations



related qualifiers

none

restrictions

none

description

This qualifier is used to alter the priority of the executing process. The priority may be lowered to reduce system overhead. The user must have ALTPRI privilege to raise priority; none is required to lower it

BacPac resets the priority when it completes execution.

example

```
$ BACPAC/PRIORITY=3 [000000...]*.*/SINCE=BACKUP MKA400:ALLDISK (1)
```

(1) An incremental backup of the entire disk will be written to the pacset ALLDISK.PAC on the MKA400: tape at a priority level of 3.



/PROTECTION

Sets the protection level on the generated tape pacset.

type

Command qualifier.

format

```
/PROTECTION=code
```

qualifier value

code

The code indicates the access level (read, write, execute and delete) for the four groups of users (system, owner, group, and world).

operations



related qualifiers

none

restrictions

none

description

This qualifier will set the protection level on the generated pacset on tape only. If not specified, no protection is set.

example

```
$ BACPAC/PROTECTION=(G:RWED)/REWIND [000000...]*.* MKA400:ALLDISK (1)
```

(1) A backup of the entire disk will be written to the pacset ALLDISK.PAC on the MKA400: tape. The indicated group protection will be written to header records of the tape. Any subsequent pacsets written to the same tape will receive the same protection code.



/RECORD

Requests that processed files have the backup date field in the file header updated to the current time.

type

Command qualifier.

format

/RECORD

/RECORD=0 (clear the backup date)

qualifier value

0 (zero)

Requests that the backup date field of the restored files be set to zero, indicating the file is not backed up.

operations



related qualifiers



restrictions

none

description

By default, BacPac does not change dates in a file header. The exception is the backup date field. /RECORD causes BacPac to record the current date-time in the backup time field of each selected file header record. This will be the time of creation of the pacset and all files marked will have the same time. This provides for saved files to be marked as backed up.

Sufficient user privilege is required.

When specified with a value of zero on a restore operation, requests that the backup date field of the newly restored files be set to zero indicating that the files have not been backed up.

examples

```
$ BACPAC/IMAGE/RECORD DKA500: MKA300:IMG500 (1)
```

(1) All files on the disk DKA500: will be saved as well as volume information. Each file will be updated to have its backup date field reflect the time of pacset generation.

The resulting pacset, IMG500.PAC, will reside on the tape mounted on device MKA300.

```
$ BACPAC/RECORD=0 MUA0:USERFILES [*...]*.* (2)
```

(2) The files contained in the pacset USERFILES.PAC will be restored. Each restored file will be updated to have its backup date field set to zero to indicate the file has never been backed up.

All restored files will reside in the original directory and have the same file name and type as the corresponding saved files.



/RELEASE_TAPE

Dismounts and unloads the tape at the completion of the save operation.

type

Command qualifier.

format**/RELEASE_TAPE****operations****related qualifiers****[/DELETE] [/RECORD]****restrictions**

none

description

On a save operation, requests that the specified tape be dismounted and unloaded at the completion of the save operation. This allows other processes to use the tape drive while BacPac completes any remaining work, e.g., /DELETE and /RECORD.

example

```
$ BACPAC/REWIND/RECORD/RELEASE_TAPE [000000...]*.* MUA0:DIRTREE (1)
```

(1) The tape in drive MUA0: will be initialized and a default label of BACPAC written to the tape. The directory tree starting at the root directory of the current default device will be written to the tape as pacset DIRTREE.PAC. At completion of writing the files to tape, the tape will be rewound, dismounted and unloaded making it available for use by another process. BacPac will continue with updating the backup date field of each file that was written to the pacset.

.....

/REMOUNT

Requests that BacPac mount the newly restored disk after an image mode duplication. The default is to mount the disk using /SYSTEM and /CLUSTER.

type

Command qualifier.

format**/REMOUNT****operations**



related qualifiers

`[/DISMOUNT]` `[/IMAGE]` `[/MOUNT_FOREIGN]`

restrictions

none

description

After completion of an image mode duplication, the target disk which was initialized and restored will be mounted for cluster and system wide use.

examples

```
$ BACPAC/DUPLICATE/DISMOUNT/MOUNT/IMAGE/REMOUNT DISK$USER1: node::DKA0:  
(1)
```

(1) The disk in drive DKA0: on node node:: will be dismounted clusterwide. The disk will then be mounted foreign before it is initialized. All files from DISK\$USER1: will be duplicated on DKA0: on node node::. At completion, BacPac will remount the disk for cluster and system use.



/REPLACE

Requests that expanded files which have matching file names replace the existing files.

type

Restore command qualifier.

format

/REPLACE

operations



related qualifiers

`[/NEW_VERSION]` `[/OVERLAY]`

restrictions

none

description

Allows a file to be replaced when an identically named file is encountered during a restore operation. BacPac deletes the existing file and processes the new file with the same version number.

By default, BacPac will not expand a file with an identical file name to one that already exists. An error message is generated. Using the /REPLACE qualifier is one way to override this default condition. /NEW_VERSION and /OVERLAY are others.

example

```
$ BACPAC MYFILES/PACSET/REPLACE *.* (1)
```

(1) The files contained in the pacset MYFILES.PAC in the current default directory will be restored. If existing files with the same version number exist for the files contained in the pacset, the saved files will be restored, replacing the existing files with the same version number. All restored files will reside in the current default directory and have the same file name and type as the corresponding saved files.

.....

/REWIND

Specifies the output tape be initialized on a save operation. Specifies that a restore or list operation start processing the tape at the beginning.

type

Command qualifier.

format

/REWIND

operations

SAVE	RESTORE	LIST	COPY

related qualifiers

[/IGNORE] [/LABEL] [/PROTECTION] [/TAPE_EXPIRATION]

restrictions

none

description

On a save operation, requests that the specified tape be rewound to the beginning of tape mark and initialized. A new label will be written to the tape and any old data will be lost.

On a restore or list operation, requests that the tape be read from the beginning of tape mark. If necessary, the tape will be rewound.

examples

```
$ BACPAC/REWIND [000000...]*.* MUA0:DIRTREE (1)
```

(1) The tape in drive MUA0: will be initialized and a default label of BACPAC written to the tape. The directory tree starting at the root directory of the current default device will be written to the tape as pacset DIRTREE.PAC.

```
$ BACPAC/LIST/REWIND MUA0:DIRTREE (2)
```

(2) The tape in drive MUA0: will be rewound to the beginning of tape. The pacset DIRTREE.PAC will be read and displayed.



/SELECT

Selects files for processing.

type

File selection qualifier.

format

```
/SELECT=select-file-spec
```

qualifier value

file specification list of the files to be excluded.

There is no default value.

operations



related qualifiers

```
[/EXCLUDE]
```

restrictions

none

description

This qualifier is used to selectively list or restore files. The select-file-spec may contain a list of file specifications. In this case, the list must be contained in parentheses. Wild cards and sticky defaults are supported. Relative version numbers are not supported.

examples

```
$ BACPAC TEST/SELECT=*.COM/PACSET *.* (1)
```

(1) The files contained in the pacset TEST.PAC which have file type .COM will be restored. All files will reside in the current default directory. The restored files will have file names, types and version numbers which match those of the corresponding saved files.

```
$ BACPAC/LIST MUA0:MYDIR/SELECT=(*.C,*.MAR) (2)
```

(2) The files contained in the pacset MYDIR.PAC which resides on the tape mounted on MUA0: and which have file type .C and file type .MAR will be listed in the brief LIST display.

.....

/SINCE

Requests that files dated later than the time specified be selected for processing.

type

Input file selection qualifier.

format

/SINCE (default is TODAY)

/SINCE=time

qualifier value

time (see text below.)

operations

SAVE	DUPLICATE
	

related qualifiers

[/BACKUP] **[/BEFORE]** **[/CREATED]** **[/DIFFERENTIAL]** **[/EXPIRED]** **[/INCREMENTAL]**
[/MODIFIED]

restrictions

none

description

Files are selected from the input-file-spec which are dated the same or later than the time specified. The time entered may be absolute time or delta time. Refer to the "OpenVMS DCL Concepts Manual" for the proper format. The time may also be selected using one of the following keywords:

- BACKUP** The backup file time in the file's header
- TODAY** The current day at 00:00:00.0 o'clock
- TOMORROW** 24 hours after 00:00:00.0 o'clock of the current day
- YESTERDAY** 24 hours before 00:00:00.0 o'clock of the current day

If no time value is specified, TODAY is assumed.

The file time selected from the file header is dependent upon the use of the qualifiers /CREATED, /EXPIRED, /MODIFIED, or /BACKUP. If none of these other qualifiers is present on the command line, /MODIFIED is assumed.

examples

```
$ BACPAC/SINCE=TODAY/MODIFIED *.FOR;* SRC/PACSET (1)
```

(1) All files in the current default directory with file type .FOR will be candidates for selection. Of the set of candidates, files will be selected which have a revision time in the file header which is later than 00:00 o'clock of the current day. The generated pacset will be called SRC.PAC; it will reside in the current default directory.

```
$ BACPAC/SINCE=BACKUP/MODIFIED [*...] MKA500:DIRTREE (2)
```

(2) All files in the directory tree starting at the root level which have a revision time in the file header which is later than the backup time for the same file will be saved. The generated pacset will be called DIRTREE.PAC; it will be written to the tape in drive MKA500:.



/STATISTICS

Request statistical information regarding the execution process be displayed.

type

Command qualifier.

format

```
/STATISTICS
```


operations**related qualifiers****restrictions**

none

description

This qualifier causes statistical information regarding the process to be displayed to SYS\$OUTPUT. The statistics include elapsed time, CPU time, number of buffered I/O's, number of direct I/O's, peak number of pages of virtual memory allocated, and number of page faults. Other statistics are operation dependent. The save operation also displays total blocks input, total compressed blocks output, and the percent reduction.

The number of pages of virtual memory used represents the amount of dynamically allocated memory, i.e., the amount of virtual memory memory, i.e., the amount of virtual memory allocated by BacPac.

example

```
$ BACPAC/STATISTICS SYS$SYSTEM:*. * MUA0:SYS (1)
```

(1) All files in the directory pointed to by the logical name SYS\$SYSTEM will be saved. At the end of the save operation, statistical information will be sent to SYS\$OUTPUT. An example of this is given below.

BacPac Statistics

```
Elapsed Time: 00:00:11.85  CPU Time: 0:00:07.33
Buffered I/O:      146 Direct I/O:    45
Virtual Memory Used: 1089 Page Faults:  381
```

```
295 blocks (18 input files) compressed to 63 blocks (78.6 percent
reduction)
```

.....

/SUMMARY

Requests a summary display of a pacset.

type

Command qualifier.

format

/SUMMARY

operations

```
| LIST |  
|-----|
```

related qualifiers

[/BRIEF] [/COMMENT] [/EXCEPTIONS] [/FULL] [/LIST] [/NONE]

restrictions

none

description

/SUMMARY provides for a summary display of the contents of one or more pacsets to SYS\$OUTPUT. The display consists of the pacset description (if input with /COMMENT during a save operation) and the version of BacPac and the operating system used. Specific information about individual files is not given.

example

```
$ BACPAC/LIST/SUMMARY MUA0:* (1)
```

A summary display of each pacset on the tape mounted in MUA0: will be produced.

.....

/TAPE_EXPIRATION

Specifies the expiration date to be written to the tape file header record. This field is checked during label processing of subsequent operations.

type

Save command qualifier.

format

/TAPE_EXPIRATION=dd-mmm-yyyy

qualifier value

dd-mmm-yyyy

Specifies the day, month and year that the tape expiration date is to be set to.

operations**related qualifiers**

`[/IGNORE]` `[/LABEL]` `[/REWIND]`

restrictions

none

description

Writes the specified date to the file header label of the tape. This is checked in future operations. Tapes which have not yet reached their expiration date may not be written to without specifically requesting that label processing be ignored (See `/IGNORE`).

examples

```
$ BACPAC/REWIND/TAPE_EXPIRATION=31-DEC-1994 [...] *.* MUA0:DIRTREE (1)
```

(1) The files contained in the directory tree starting at the current default directory will be written to the pacset DIRTREE.PAC which will be written to the tape in MUA0:. The tape is to be initialized (`/REWIND`) and an expiration date of Dec 12, 1994 written to the header record.

```
$ BACPAC/COPY/TAPE_EXPIRATION=31-DEC-2020 MUA0: MKA300: (2)
```

(2) The tape in device MUA0: will be copied to the tape in device MKA300:. The tape label will be the same on the output tape as the input tape. The expiration date of the copied output tape will be set to the future time.

.....

/TRUNCATE

Requests that restored files be truncated at the end-of-file.

type

Restore command qualifier.

format

`/TRUNCATE`

operations

related qualifiers

`[/OVERLAY]`

restrictions

none

description

/TRUNCATE requests that the restored file(s) be truncated at the end-of-file. By default, the allocation of the original file determines the size of the restored file.

If the pacset is restored to a different device, the allocation of the restored file may be affected by the device cluster size (see "Guide to OpenVMS System Management and Daily Operations") of the new device. Use of the /TRUNCATE qualifier may reduce the amount of disk space allocated for the expanded files if the new device cluster size is different from that of the original device.

note

/TRUNCATE may only be used for sequential files. If specified for indexed or relative files, an informational message will be generated and the qualifier will be ignored for the file in question.

example

```
$ BACPAC TEST/TRUNCATE/PACSET *.* (1)
```

(1) The pacset TEST.PAC in the current default directory is to be restored. The allocation of each restored file will be set to the end-of-file block. The actual allocation will depend on the device cluster size of the system on which the files are being expanded.

All restored files will reside in the current default directory. Their file names, types and version numbers will match those of the original files.



/VERIFY

Requests that files be expanded in memory with CRC checking.

type

Command qualifier.

format

/VERIFY

/NOVERIFY (default)

operations



related qualifiers

`[/LIST]` `[/LOG]` `[/NONE]`

restrictions

none

description

This qualifier requests verification of the pacset.

An additional pass is made at the end of the save operation. The pacset is read and each of the newly saved files is expanded in memory to verify that the CRC generated for the expanded file matches the CRC of the original file. An error message is output if the CRCs do not match. Use of this qualifier will verify that the contents of the pacset can be restored at a later time. If /LOG has been specified, a message is output for each file as it is verified.

If /VERIFY is used with /LIST, as each saved file is read it is expanded in memory with checking of CRCs. An error message is output if the CRCs do not match. If /MONITOR is also requested, a message indicating successful verification is output as each file is verified.

examples

```
$ BACPAC/DELETE/VERIFY *.COM COMFILES/PACSET (1)
```

(1) All files in the current directory with extension .COM will be saved. Once the pacset has been created, the verification pass will be done. If any errors occur during this pass, no files will be deleted. If the verification pass is successful, the input files will then be deleted.

```
$ BACPAC/LIST/NONE/VERIFY MUA0:* (2)
```

(2) Each pacset on the tape mounted on MUA0: will be verified. All files will be expanded in memory with CRC checking done. Error messages will be displayed for any files which do not verify. No other output will be given.

.....

/VERSION

Displays the version of BacPac.

type

Command qualifier.

format

/VERSION

operations



related qualifiers

none

restrictions

cannot be used with any other qualifier

description

Displays the version of BacPac to SYS\$OUTPUT. The level indicates the internal structure revision level of the PacSets created with this version of the software. All previous levels are supported for restore operations, however, only files with the level indicated will be created.

example

```
$ BACPAC/VERSION (1)
```

(1) The following information will be displayed to SYS\$OUTPUT.

```
BacPac Version 5.6-0 Evaluation Kit Level D  
Copyright 1994-2006 Compact Data Works, Inc.  
All rights reserved.  
Licensee: EVALUATION KIT  
Serial Number: 000000  
Node ID: xxxxxxxx  
License Type: Evaluation
```



/VOLUME_NAME

Specifies that BacPac use the specified volume name for the newly restored disk.

type

Command qualifier.

format

```
/VOLUME_NAME=volume_name
```

qualifier value

character string for the volume name of the restored disk.

operations**related qualifiers**

`[/INITIALIZE][/NAME]`

restrictions

none

description

After an image mode duplication, the target disk may reside on another node or may be part of the same cluster as the source disk. The two disks will have the same volume name and, if part of a cluster, may not both be mounted. `/VOLUME_NAME` provides a way to have two identical disks be part of the same cluster. They differ only in the volume name and thus may both be mounted at the same time.

examples

```
$ BACPAC/DUP/VOLUME_LABEL=DISK2 DKA0: ALPHA::DKA100: (1)
```

(1) The local disk DKA0 is to duplicated onto the disk DKA100: on node ALPHA. The volume name of the disk DKA100: will be 'DISK2'. Other than the volume name, the 2 (DKA0 on the local node and DKA100 on node ALPHA) disks will be virtually the same.

**/ZERO_DATA**

Write zeros to files which have not had their data saved.

type

Restore command qualifier.

format

`/ZERO_DATA`

operations**related qualifiers**

none

restrictions

none

description

Files which have not had their data saved, e.g., files which are marked 'Backups Disabled' have blocks allocated to them which may be written to with zeros by using this qualifier. This is strictly a security operation.

example

```
$ BACPAC/ZERO_DATA MUA0: DUA0: (1)
```

(1) The first save set on the tape mounted in MUA0: will be restored to the DUA0: disk. Any files which have not had their data saved, will be zeroed to the end-of-file marker.



Index

/	
/ALIAS	47
/ASSIST.....	42, 43, 48
/BACKUP.....	49
/BEFORE.....	51
/BLOCK_SIZE	52
/BRIEF	53
/BROADCAST	27, 54
/BY_OWNER.....	55
/CBT	23, 56
/COMMENT.....	57
/COMPRESSION	58
/CONFIRM.....	27, 59
/CONTIGUOUS.....	23, 60
/COPY	43, 61
/CREATED	62
/DELETE	63
/DENSITY	64
/DIFFERENTIAL.....	29, 30, 65
/DISMOUNT	66
/DUPLICATE	67
/ENCRYPTION.....	68
/EXCEPTIONS	70
/EXCLUDE	23, 71
/EXPIRED.....	72
/FULL	73
/IGNORE.....	27, 41, 42, 43, 74
/IMAGE	27, 76
/INCREMENTAL	22, 29, 30, 77
/INITIALIZE	27, 78
/JOURNAL	30, 79
/KEEP	80
/LABEL.....	41, 43, 81
/LIST	83
/LOCK_DEVICE.....	27, 84
/LOG	85
/MEDIA_FORMAT	86
/MODIFIED	87
/MOUNT_FOREIGN	88
/NAME.....	89
/NEW_VERSION	23, 90
/NONE.....	91
/NOTIFY.....	27, 92
/OVERLAY	23, 93
/OWNER_UIC	55
/PACSET.....	32, 33, 94
/PASSWORD	42, 95
/PRIORITY	25, 96
/PROTECTION	97
/RECORD	22, 29, 97

/RELEASE_TAPE99

/REMOUNT 100

/REPLACE.....23, 100

/REWIND.....27, 41, 101

/SELECT23, 102

/SINCE22, 29, 103

/STATISTICS 105

/SUMMARY.....32, 106

/TAPE_EXPIRATION.....43, 107

/TRUNCATE.....23, 108

/VERIFY 109

/VERSION 110

/VOLUME_NAME..... 111

/ZERO_DATA..... 112

B

Batch Files.....24

C

CRC.....21, 91, 109

CTRL-C..... 21, 84

CTRL-T 25

CTRL-Y 84

D

DECnet 31

E

Encryption 39, 68

I

Image Mode 27

J

Journal Files..... 20, 21, 22

L

Logical names 41, 81

O

OPCOM Messages 42, 43

U

UIC..... 42