

# **KnapSac Client User's Guide**

---



# Table of Contents

KnapSac for OpenVMS and Windows or Linux .....	3
Introduction .....	3
Using this Help .....	3
Online Help .....	4
Client Installation on VMS .....	5
Installation Requirements .....	5
Installation Options .....	5
Selecting an Installation Directory .....	5
Conventions used during Installation .....	6
Installation Procedure .....	6
Installing KnapSac as a known image .....	8
KnapSac License File .....	8
Updating a License file .....	9
KnapSac Operations .....	11
Save Operations .....	11
Restore Operations .....	11
List Operations .....	12
Directory Operation .....	12
Support .....	13
Using the KnapSac Server .....	15
Port Numbers .....	15
Client Path .....	15
Server Scenarios .....	15
Getting Started .....	17
Verifying KnapSac Client is Installed and Available .....	17

Creating a Pacset .....	17
Listing a Pacset .....	18
Selecting files to be saved .....	18
Restoring a Pacset .....	19
Invoking KnapSac from Within Batch Files .....	19
Getting a Directory of a Windows System .....	20
Translating an Exit Code .....	20
Common KnapSac VMS Exit Status Codes .....	20
Using CTRL-T to Monitor Progress .....	21
Setting the Process Priority Using /PRIORITY .....	21
Image Mode KnapSac.....	23
Image Mode Save .....	23
Image Mode Restore .....	23
Incremental and Differential Backups .....	25
Incremental Save.....	25
Differential Save .....	25
Incremental Restore .....	26
Differential Restore.....	26
Encryption .....	27
Encryption Keys and Algorithms.....	27
Using Encryption .....	27
Qualifiers .....	29
Qualifiers Reference.....	29
ALIAS.....	29
BACKUP .....	30
BEFORE .....	32

BRIEF .....	33
BROADCAST .....	34
BY_OWNER .....	35
CBT.....	36
COMMENT .....	37
COMPRESSION.....	38
CONFIRM.....	39
CONTIGUOUS .....	40
CREATED .....	41
DELETE.....	42
DIFFERENTIAL .....	43
ENCRYPT .....	44
EXCEPTIONS .....	46
DIR.....	46
EXCLUDE.....	47
EXPIRED.....	48
FULL.....	50
IGNORE.....	51
IMAGE .....	52
INCREMENTAL.....	53
INITIALIZE.....	54
JOURNAL .....	55
LIST .....	57
LOCK_DEVICE .....	58
LOG .....	59
MODIFIED.....	60

NEW_VERSION .....	61
NONE .....	62
NOTIFY.....	63
OVERLAY.....	64
PORT.....	65
PASSWORD.....	66
PRIORITY.....	67
RECORD.....	68
REPLACE.....	69
SELECT.....	70
SINCE.....	71
STATISTICS.....	72
SUMMARY .....	73
SUBDIRS.....	74
TRUNCATE.....	75
VERIFY.....	76
VERSION .....	77
ZERO_DATA.....	77

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 and KnapSac are trademarks of Compact Data Works, Inc..

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

Windows is a registered trademark of Microsoft Corporation.



Copyright 2006-2014 Compact Data Works, Inc.

All Rights Reserved



# KnapSac for OpenVMS and Windows or Linux

## Introduction

KnapSac is a powerful and fast backup utility which combines state of the art file compression techniques, client/server architecture, network communications, and extensive file handling operations to store VMS backup data on Windows computers where the cost of disk space is considerably less.

KnapSac utilizes a client-server architecture where the client software resides on a VMS computer and the server software resides on a Windows or Linux computer. The client and server communicate using TCP/IP.

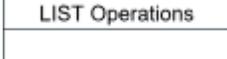
KnapSac generates saveset files similar to those generated by the VMS BACKUP Utility. These files are called pacsets. Files restored from a pacset will exactly match the original file in data content and file structure.



## Using this Help

This Help system contains the entire KnapSac Reference Manual.

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 'Locate' button (Windows) or 'Sync' button (Linux) at the top of the Table of Contents will show you where the current topic is. The 'Hide' button (windows) or the re X (Linux) will 'hide' the Table of Contents. To bring it back, simply click the 'Show' button (Windows) or one of the buttons in the help header (Linux).

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 KnapSac client installation kit. During installation, it is added to the VMS help library and is accessible with the following VMS command:

**\$ help knapsac**

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.



# Client Installation on VMS

## Installation Requirements

The installation kit consists of one distribution volume labeled AXP KNS version k.n-m or IA64 KNS 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 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 7.2 or higher (Alpha)

VMS Version 8.2 or higher (Integrity)

## Installation Options

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



## Selecting an Installation Directory

The installation of KnapSac requires one directory. The logical name KS\_Manager is defined to point to this directory. Examples of files which reside in this directory are:

- A copy of the KNAPSAC.CLD file which defines the KnapSac command.
- The KnapSac HELP if it is not integrated with VMS HELP.
- The KnapSac client executable image files.

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

During the installation, a list of system files which are new or modified by the installation procedure is displayed. If the KnapSac 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 KnapSac command. If the command is not integrated with the system DCL tables, any user wishing to use KnapSac must first execute the following command after logging in:

```
$ SET COMMAND KS_MANAGER:KNAPSAC
```

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

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 KnapSac 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:KNAPSAC_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:

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 KnapSac:

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.

\* Where will the distribution volumes be mounted:

6. Enter IA64KNS or AXPKNS for the product name and confirm that you are ready.

\* Products: **AXPKNS**

\* 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 KnapSac files. If you answer YES, the procedure will create the system level directory SYS\$SYSROOT:[KNAPSAC]. 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 KS\_MANAGER will be defined to point to the KnapSac directory.

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

\* Enter the device and directory for KnapSac installation:

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

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

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

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

11. You will now be asked if you already have a KnapSac license installed. If you are installing an update to KnapSac and you already have a license file, answer YES. If you are installing KnapSac 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 KnapSac license already installed [NO]?

\* Enter the KnapSac evaluation kit key: XXXXXX

12. 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.

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

.....

## Installing KnapSac as a known image

You may want to install KnapSac 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 KnapSac:

```
$ INSTALL
```

```
INSTALL> ADD/OPEN/HEADER KNAPSAC
```

```
INSTALL> EXIT
```

These commands may be inserted in the KNAPSAC\_STARTUP.COM command file in SYS\$MANAGER or alternatively added to the normal VMS startup command file. If you insert these commands in KNAPSAC\_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.

.....

## KnapSac License File

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

The license file created by the installation procedure are of limited time duration. Once the license file expired, KnapSac will cease to function.

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

```
$ knapsac/version
```

The version of KnapSac 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 KS\_LICENSE.COM.

```
$ @KS_MANAGER:KS_LICENSE
```

Print the node IDs that are generated by the procedure and email 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 KnapSac an expiration date will no longer appear.

.....



# KnapSac Operations

## Save Operations

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

### Format

**KnapSac 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.

#### output-destination

The output-destination is a Windows or Linux computer. The node name is required followed by a Windows or Linux file specification which is inclosed in quotation marks.

### Qualifiers

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

[/ALIAS] [/BACKUP] [/BEFORE] [/BROADCAST] [/COMMENT] [/COMPRESSION] [/CONFIRM]  
[/CREATED] [/DELETE] [/DIFFERENTIAL] [/ENCRYPT] [/EXCLUDE] [/EXPIRED]  
[/IGNORE] [/IMAGE] [/INCREMENTAL] [/JOURNAL] [/LIST] [/LOCK\_DEVICE] [/LOG] [/MODIFIED]  
[/NOTIFY] [/PACSET] [/PASSWORD] [/PRIORITY] [/RECORD] [/SINCE] [/STATISTICS] [/VERIFY]

## Restore Operations

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

### Format

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

### Parameters

### input-pacset

The input-pacset specifies the Windows or Linux node name followed by a Windows or Linux file specification enclosed in quotation marks.

### 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 path 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.

```
[/BROADCAST] [/BY_OWNER] [/CBT] [/CONFIRM] [/CONTIGUOUS] [/ENCRYPT] [/EXCLUDE]
[/IGNORE] [/IMAGE] [/INCREMENTAL] [/INITIALIZE] [/LIST] [/LOG] [/NEW_VERSION]
[/NOTIFY] [/OVERLAY] [/PACSET] [/PRIORITY] [/RECORD] [/REPLACE]
[/SELECT] [/STATISTICS] [/TRUNCATE] [/ZERO_DATA]
```

## List Operations

KnapSac can list the contents of pacsets which reside on a Windows system.

### Format

```
KnapSac/list input-pacset [/qualifier1.../qualifiern]
```

### Parameters

#### input-pacset

The input-pacset specifies the Windows or Linux node name followed by a Windows or Linux file specification enclosed in quotation marks.

### Qualifiers

Many qualifiers are available for list operations. The only required qualifier is /LIST.

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

## Directory Operation

KnapSac provides a VMS style directory listing of a folder on Windows. This provides visibility into the pacsets that have been generated.

### Format

`knapsac/directory input-file-spec [/qualifier1.../qualifiern]`

### Parameters

`input-file-spec`

The `input-file-spec` specifies the Windows node name followed by a Windows file specification enclosed in quotation marks.

### Qualifiers

The only required qualifier is `/DIR`.

`[/DIR] [/SUBDIRS]`

### Related Topics

## Support

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

### Format

`knapsac/version`

### Qualifiers

`[/VERSION]`



# Using the KnapSac Server

## Port Numbers

The KnapSac client on VMS communicates with the KnapSac server on Windows/Linux using TCP.

The KnapSac server on Windows/Linux is configured to run one or more server processes. Each KnapSac server process can handle up to 10 consecutive TCP port numbers which means it can handle up to 10 concurrent connections from a KnapSac client using a single process thus reducing overhead on the server computer. The number of KnapSac server processes and which TCP port numbers each KnapSac server process handles is specified in the KnapSac Configuration file on the Windows/Linux computer. The configuration file also specifies the top level path each server process uses when storing or retrieving files on the Windows/Linux computers.

When a KnapSac server process is activated, it listens on the TCP port numbers specified in the KnapSac configuration file. Normally, the first of the 10 port numbers is also specified in the VMS KnapSac license file. This is the default port number used for a connection. To access one of the other port numbers, the VMS client command must contain the /PORT qualifier to specify a particular port number.

```
$ knapsac *.dat server::datfiles.pac /port=7005
```

.....

## Client Path

When the KnapSac server is configured on a Windows or Linux system, a top level path is established for each TCP port number that will be used to provide communications between the VMS client and the KnapSac server. For example, suppose the KnapSac server is configured to use the path

```
c:\vmspacsets
```

for storing or retrieving pacsets associated with port 7000. Then the KnapSac client command

```
$ knapsac *.dat server::datfiles.pac
```

will store the pacset datfiles.pac in the folder c:\vmspacsets\. The KnapSac client command

```
$ knapsac *.dat server::"15may09\datfiles.pac
```

will store the pacset in the folder c:\vmspacsets\15may09\. If the sub-directory 15may09 does not exist, it will be created. The quote marks are necessary in this latest example.

.....

## Server Scenarios

A KnapSac Server Configuration can be a simple one using a single server process or it can be a complex one consisting of many server processes.

In the simple scenario, a single server process is used to support up to 10 consecutive ports. For example, a single VMS system with multiple drives can be backed up to a single server process using a unique port for each disk drive. This allows up to 10 disks to be backed up concurrently.

If there are multiple VMS computers, a separate server process can be used for each VMS client, allowing multiple disks on multiple VMS computers to be backed up concurrently. And since each server process can have a unique storage path, the pacsets from the different VMS computers will be kept separate.



# Getting Started

## Verifying KnapSac Client is Installed and Available

Once KnapSac is installed on your system, you may need to issue a **SET COMMAND** to access the KnapSac command; this is dependent on the type of installation selected (See the Installation Instructions). The following command will tell you whether KnapSac 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.

```
$ KNAPSAC/VERSION
```

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

```
$ SET COMMAND KS_MANAGER:KNAPSAC
```

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 only be created on a Windows or Linux system. The command and related qualifiers are very similar to those used by BacPac for VMS. You may be able to use the same command procedures you are currently using by simply changing the verb from BACPAC to KNAPSAC and modifying where the pacset resides.

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

```
$ knapsac *.* win::mydir
```

To create the same pacset on Linux,

```
$ knapsac *.* linux::mydir
```

This command will create a pacset containing all the files in your current default directory. The pacset created will be called mydir.pac. The default filename extension for any pacset generated by KnapSac is .pac. You may override this simply by explicitly giving an extension on the command line (mydir.pac or mydir.bck).

KnapSac provides an alternate syntax which you may find easier to use. You specify the operation you desire (**SAVE**, **RESTORE**, or **LIST**). The above command alternatively could have been:

```
$ knapsac save *.* win::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:

```
$ knapsac *.*;0 win::mydir
```

or

```
$ knapsac save *.*;0 linux::mydir
```

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



## Listing a Pacset

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

```
$ knapsac/list win::mydir
```

The alternate syntax for this command would be as follows:

```
$ knapsac list linux::mydir
```

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.

```
$ knapsac/list/verify linux::mydir
```

This command requests that each file be checked for data integrity by verifying it's CRC.



## 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.

```
$ knapsac *.* /since=today/created win::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.

```
$ knapsac *.*;0/since=backup linux::new
```

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 KnapSac to set the BACKUP date field of each file header.



## Restoring a Pacset

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

```
$ knapsac win::mydir *.*
```

or using the alternate syntax,

```
$ knapsac restore win::mydir *.*
```

will restore the pacset created in the first example. If the files exist in the current directory, they will not be restored. By default, KnapSac 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.

```
$ knapsac linux::mydir [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.

```
$ knapsac linux::mydirtree [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 KnapSac from Within Batch Files

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

```
$ SET COMMAND KS_MANAGER:KNAPSAC.CLD
```

KnapSac can be invoked from within DCL command files which are executed interactively or as batch jobs. As with most VMS utilities, KnapSac generates a condition code when it completes execution. If KnapSac completes successfully, the condition code will have a true (odd) value. If KnapSac 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.



## Getting a Directory of a Windows System

KnapSac can display a VMS style directory listing of files on a Windows system. The display lists the filenames and file sizes in VMS blocks. For example,

```
knapsac /dir win::
```

will list the filenames in the drive and folder pointed to by the Storage Path. To list only the files with extension .pac, use the following

```
knapsac /dir win:: *.pac
```

To list all of the sub folders, use the following

```
knapsac /subdir win::
```

This will list the contents of all the folders starting at the one pointed to by the Storage Path.

## Translating an Exit Code

If you are running KnapSac from within a batch file or called from another program, there may be times when you wish to translate an exit code. KnapSac messages are provided with the distribution kit and reside in the KS\_MANAGER directory. To translate a given exit code, execute the following DCL commands:

```
$ SET MESSAGE KS_MANAGER:KNAPSAC_MSG  
$ symbol = F$MESSAGE(exit_code)  
$ SHOW SYMBOL symbol
```

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

## Common KnapSac VMS Exit Status Codes

Event or Condition	VMS Exit Status Code	Description or Probable Cause
--------------------	----------------------	-------------------------------

Timeout while connecting to the Server	%X1000022C	Server is not running or the Client and Server are using different port numbers
Link disconnected during a copy operation	%X100020DC	Link went down or the Server stopped
File could not be created	%X1E9BA10C	File already exists or protection violation
Directory cannot be created	%X1E9BA114	Unknown output device (drive letter not mapped)
Error writing file	%X1E9BA0FC	Insufficient disk space or disk error
Connect to network object rejected	%X10000294	The Server is not running
Unknown host	%X1E9B9FEC	The Server node is not in the VMS TCP database
Successful operation	%X00000001	Operation completed without errors



## Using CTRL-T to Monitor Progress

During any KnapSac operation, KnapSac 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 KnapSac is currently processing. This is most useful if you are saving entire disks and would like to see 'progress reports' as the operation proceeds. KnapSac 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 KnapSac interactively, or even from a batch job, you may wish to lower the priority of the process executing KnapSac. This is accomplished easily using the /PRIORITY qualifier. The following example will save a file and set the process priority to 3.

```
$ KNAPSAC/PRI=3 LARGE.DAT WIN: :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 KnapSac

## Image Mode Save

The simplest and the most efficient backup is an image mode save operation. This operation will write the contents of an entire disk to a pacset on a Windows or Linux system. 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, input file specifications are not allowed.

```
$ KNAPSAC/IMAGE DISK$USER: WIN::usersave
```

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 in the backups directory on the C drive on the Windows system WIN.

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'.

```
$ KNAPSAC *.* WIN::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 create a standalone backup. 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 KnapSac 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 KnapSac is done. If errors occurred, KnapSac will mail you the error log (PACERROR.LOG). If no errors occurred, KnapSac will simply tell you this. You may use the /NOTIFY qualifier with any KnapSac 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. KnapSac will initialize the disk and then restore all files from the image mode pacset.

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

```
$ DISMOUNT DKA0:  
$ MOUNT/FOREIGN DKA0:  
$ KNAPSAC/IMAGE WIN::dk0save DKA0:
```

This set of commands will completely restore the disk DKA0: from the pacset dk0save.pac in the backups directory on the C drive of the Windows system WIN.. 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 KnapSac.

```
$ DISMOUNT DKA0:  
$ INITIALIZE/HEADERS=10028 DKA0: HENRY  
$ MOUNT/FOREIGN DKA0:  
$ KNAPSAC/IMAGE/NOINITIALIZE WIN::dk0save DKA0:
```

These commands will first initialize the disk, give it a volume name of HENRY and create 10028 file headers. KnapSac 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 KnapSac 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.



# 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.

```
$ knapsac/since=backup /record disk$user:[000000...]*.*;*-  
$_ win::userdisk
```

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.

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

```
$ knapsac/incremental disk$user: win::userdisk
```

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 KnapSac and VMS 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.



## 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.

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

```
$ knapsac/differential disk$user: win::userdisk
```

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. KnapSac 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, KnapSac will default it to KNAPSAC.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:
$ knapsac/image/record win::dua0.pac dua0:
$ dismount dua0:
$ mount dua0: diskname
$ knapsac/incremental win::update1.pac dua0:
$ knapsac/incremental win::update2.pac dua0:
$ knapsac/incremental win::update3.pac 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:
$ knapsac/image/record win::dua0.pac dua0:
$ dismount dua0:
$ mount dua0: diskname
$ knapsac win::update3.pac dua0:
```

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



# Encryption

## Encryption Keys and Algorithms

KnapSac 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 KnapSac 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, KnapSac 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 KnapSac commands.

.....

## Using Encryption

KnapSac supports AES encryption through the use of the qualifier /ENCRYPT.

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

```
$ knapsac *.* win::mydir /encrypt=(name=keyname)
```

This command will create a pacset containing all the files in your current default directory. The key used for encryption has been previously defined using the VMS ENCRYPT command.

Internally, KnapSac 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:

```
$ knapsac win::mydir /encrypt=(name=keyname) *.*
```

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

```
$ knapsac *.* win::mydir /encrypt
```

.....

# Qualifiers

## Qualifiers Reference

Each of the available KnapSac 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][/BACKUP][/BEFORE][/BRIEF][/BROADCAST][/BY_OWNER][/CBT][/COMMENT]`  
`[/COMPRESSION][/CONFIRM][/CONTIGUOUS][/CREATED]`  
`[/DELETE][/DIFFERENTIAL][/DIR][/ENCRYPT][/EXCEPTIONS][/EXCLUDE][/EXPIRED][/FULL]`  
`[/IGNORE][/IMAGE][/INCREMENTAL][/INITIALIZE][/JOURNAL][/LOCK_DEVICE][/LOG][/MODIFIED]`  
`[/NEW_VERSION][/NONE][/NOTIFY][/OVERLAY]`  
`[/PASSWORD][/PORT][/PRIORITY][/RECORD][/REPLACE][/SELECT][/SINCE][/STATISTICS]`  
`[/SUBDIRS][/SUMMARY][/TRUNCATE][/VERIFY][/VERSION][/ZERO_DATA]`

### /ALIAS

Requests file processing similar to image mode.

#### type

Command qualifier.

#### format

**/ALIAS** (default)

**/NOALIAS**

#### operations

SAVE


#### 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

```
$ knapsac/noalias/since=today [000000...]*.*-  
$_win::dirtree.pac /pacset (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 Windows system WIN. 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.



## /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



### 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 KnapSac 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

```
$ knapsac *.doc/backup/before win::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.

```
$ knapsac *.mem/backup/since=12-MAY-1995 win::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.

```
$ knapsac *.mem;*/nobackup/before=yesterday win::memfiles (3)
```

(3) All files in the current default directory which have file type .MEM and which have been marked as <No backup done> will be selected for processing. Files which have a backup date prior to 'yesterday' will also be selected. The resultant pacset will reside on the Windows system WIN and will be called memfiles.pac.

```
$ knapsac *.* /ignore=nobackup win::bckfiles (4)
```

(4) 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



### 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.

- |                 |  |
|-----------------|--|
| <b>BACKUP</b>   | The backup file time in the file's header            |
| <b>TODAY</b>    | The current day at 00:00:00.0 o'clock                |
| <b>TOMORROW</b> | 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

```
$ knapsac *.* /before win::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 Windows system WIN. It will be called mydir.pac.

```
$ knapsac *.*;*/before=10:00 win::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 called mydir.pac.

```
$ knapsac *.* /before=backup/modified win::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.

.....

## /BRIEF

Requests a brief listing of a pacset.

### type

Command qualifier.

### format

**/BRIEF**

### operations

LIST
<b>■</b>

### 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 KnapSac 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**

```
$ knapsac/list/brief win::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 KnapSac image mode operation.

**type**

Command qualifier.

**format**

```
/BROADCAST  
/BROADCAST=text  
/BROADCAST="text"
```

**qualifier value**

**string** (See text below.)

**operations**



**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 KnapSac 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

```
$ knapsac/lock/image disk$user: win::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

### related qualifiers

none

### restrictions

none

### description

By default, KnapSac 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

```
$ knapsac win::docfiles /by_owner=[user] *.* (1)
```

(1) The files contained in the pacset DOCFILES.PAC 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, KnapSac 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

```
$ knapsac win::doc /cbt *.* (1)
```

(1) The pacset DOC.PAC on the Windows system WIN 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

```
$ knapsac/comment="this is a test" test.* win::test (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.

```
$ knapsac/comment=tuesday [000000...]*.* win::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 Windows system WIN.



## /COMPRESSION

Selects the compression algorithm.

### type

Command qualifier.

### format

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

### qualifier value

**FAST**

**MAXIMUM**

**NONE**

Selects the type of compression to be used.

### operations



### related qualifiers

none

### restrictions

none

### description

The /COMPRESSION qualifier selects the type of compression. On an AXP or IA64 system 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

```
$ knapsac/comp=max/image disk$user: win::user.img (1)
```

(1) An image save of the disk DISK\$USER will be performed using maximum compression. The resulting pacset will be called user.img and will reside on the Windows system WIN.

```
$ knapsac/comp=none/image disk$user: win::user.img (2)
```

(2) An image save of the disk DISK\$USER will be performed using no compression. The resulting pacset will be called user.img and will reside on the Windows system WIN.

.....

## /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

```
$ knapsac/image/confirm win::imagesave dka0: (1)
```

(1) A prompt will be issued giving the volume name of the disk DKA0:, as well as the pacset name. The user may abort the command prior to the disk being initialized, if any items are incorrect.

```
$ knapsac/confirm win::doc *.* (2)
```

(2) The files contained in the pacset doc.pac on the Windows system WIN 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****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, KnapSac 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 KnapSac 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**

```
$ knapsac win::doc /contiguous *.* (1)
```

(1) The pacset doc.pac on the Windows system WIN 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.

**/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

```
$ knapsac [...]*/*/before/created win::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 Windows system WIN.

```
$ knapsac *.doc/created/since=yesterday win::docs (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.



## **/DELETE**

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

## type

Save command qualifier.

## format

**/DELETE**

## operations



### related qualifiers



### 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.

### example

```
$ knapsac/log *.doc/delete win::docfiles (1)
```

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

After successful creation of the pacset, and if no errors were 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 the save, KnapSac will terminate at the end of the save pass; no input files will be deleted.



## /DIFFERENTIAL

Requests that KnapSac perform a differential save operation.

### type

Command qualifier.

### format

**/DIFFERENTIAL**

### operations



### related qualifiers

`[/IMAGE][/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.

### example

```
$ knapsac/differential disk$user1: win::user1_inc (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.



## /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 KnapSac by using the VMS ENCRYPT command. If the key value is specified, it must be enclosed in quotation marks.

KnapSac 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.

KnapSac 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. KnapSac 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

**\$ knapsac/encrypt=(name=keyname)/image disk\$user: win::user.img (1)**

(1) An image save of the disk DISK\$USER will be performed using the encryption key keyname which was previously defined using the VMS ENCRYPT command. The resulting pacset will be called user.img and will reside on the Windows system WIN.

```
$ knapsac/encrypt=(name=keyname,algo=aesofb128)/image disk$user: win::user.img (2)
```

(2) The same image save is done using a different encryption algorithm.



## /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

```
$ knapsac/list/exceptions win::mydir (1)
```

Status messages will be displayed.



## /DIR

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

**type**

Command qualifier.

**format**

**/DIR**

**/DIR=list-file-spec**

**qualifier value**

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

**operations**

DIRECTORY


**related qualifiers**

**[/SUBDIRS]**

**restrictions**

none

**description**

Displays the contents of a Windows folder. 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. 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**

```
$ knapsac/dir win::tapelabel (1)
```

(1) The contents of the folder c:\backups\tapelabel will be displayed.

.....

**/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**



**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**

```
$ knapsac *.*;*/exclude=(*.com,*.dat) win::myfiles (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 on the Windows system WIN.



## **/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


**related qualifiers**

<b>/BACKUP</b>	<b>/BEFORE</b>	<b>/CREATED</b>	<b>/MODIFIED</b>	<b>/SINCE</b>
----------------	----------------	-----------------	------------------	---------------

**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**

```
$ knapsac *.* /before/expired win::mydir (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 on the Windows system WIN.

```
$ knapsac *.*;*/since=today/expired win::mydir (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.

```
$ knapsac *.*;*/noexpired win::mydir (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 on the Windows system WIN.



## /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 KnapSac 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

```
$ knapsac/list/full win::fulldisk (1)
```

(1) Detailed information will be displayed to SYS\$OUTPUT;.

## /IGNORE

Requests that KnapSac ignore certain file processing restrictions.

### type

Command qualifier.

### format

**/IGNORE=option**

### qualifier value

option (See text below.)

### operations

SAVE	RESTORE
	

### related qualifiers

**[/BACKUP]**

### restrictions

none

### description

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

#### **INTERLOCK**

By default, KnapSac 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.

#### **NOBACKUP**

By default, KnapSac 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

```
$ knapsac [000000...]/ignore=nobackup win::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.

```
$ knapsac/image/ignore=inter disk$user1: win::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 Windows system WIN.



## /IMAGE

Requests that KnapSac 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]** **[/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, KnapSac 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.

## examples

```
$ knapsac/image disk$user1:/ignore=interlock win::user1_img
```

(1) All files on the disk `DISK$USER1` will be saved, including those which are open by another user. The `/IGNORE=INTERLOCK` qualifier is specified although it is set by default for image mode. Those files which are marked 'Backups disabled' will have their file headers only included. All pertinent volume information will be saved. The pacset will be called `user1_img.pac` and will be written to the Windows system `WIN`.

```
$ knapsac/image/log/confirm win::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 and the name of the pacset 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 KnapSac perform an incremental save or restore operation.

### type

Command qualifier.

### format

**/INCREMENTAL**

### operations

SAVE	RESTORE
	

### related qualifiers

`[/DIFFERENTIAL]` `[/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` is also an implied qualifier.

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.

## examples

```
$ knapsac/since=backup/incremental disk$user1: -
```

```
$ _ win::user1_inc/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 KnapSac operation. The record pass is much faster using `/INCREMENTAL` than with a similar command.

```
$ knapsac/incremental win::user1_inc disk$user1: (2)
```

(2) The files contained in the incremental pacset will be restored. 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 KnapSac initialize the disk using parameters from the image mode pacset during an image mode restore operation.

### type

Command qualifier.

### format

```
/INITIALIZE (default)
```

```
/NOINITIALIZE
```

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

```
$ knapsac/image win::user1_img disk$user1:/init (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.

```
$ knapsac/image win::user1_img disk$user1:/noinit (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 KnapSac save operations.

**type**

Command qualifier.

**format****/JOURNAL**

## **/JOURNAL=journal-file-spec**

### qualifier value

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

### operations



### 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.

KnapSac 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 KNAPSAC.PJL.

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

### examples

```
$ knapsac/journal [...]*. * win::dirtree (1)
```

(1) The directory tree starting at the current default directory will be written to the pacset dirtree.pac: The pacset name as well as the names of all files contained in the pacset will be written to the journal file KNAPSAC.PJL in the current default directory.

```
$ knapsac/list/journal (2)
```

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



## /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] [/JOURNAL] [/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. 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

```
$ knapsac/list win::dirtree (1)
```

(1) The contents of the pacset dirtree.pac will be displayed.

```
$ knapsac/list=dirtree win::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.

```
$ knapsac/list=dirtree/log [...]*. * win::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**

**/NOLOCK\_DEVICE** (default)

### 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

```
$ knapsac/lock_device/image/broadcast DUA1: win::dual_img (1)
```

(1) The user disk DUA1: will be locked against write access during the image mode save. 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**

**/NOLOG** (default)

### operations

SAVE	RESTORE	LIST
		

### 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

```
$ knapsac/log [...]*. * win::dirtree (1)
```

(1) The directory tree starting with the current default directory will be saved in the pacset dirtree.pac on the Windows system WIN. The following is a sample of the information which is sent to SYS\$OUTPUT during execution of this command:

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

.....

## /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

```
$ knapsac [...] *.*;0/before/modified win::mydir (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 on the Windows system WIN.

```
$ knapsac *.* /since=(5-JAN-2006:9:00) /mod win::dirsave (2)
```

(2) All files in the current default directory which have a revision time later than 09:00 o'clock of January 5, 2006 will be saved. The pacset will be called dirsave.pac and will reside on the Windows system WIN.



## /NEW\_VERSION

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

### type

Restore command qualifier.

### format

**/NEW\_VERSION**

### operations



### related qualifiers

**[/OVERLAY][/REPLACE]**

### 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, KnapSac 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	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**

```
$ knapsac win::myfiles /new_version *.* (1)
```

(1) The files contained in the pacset myfiles.pac from the Windows system WIN 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**



**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**

```
$ knapsac/list/none/verify win::mydir (1)
```

(1) All files in the pacset mydir.pac, residing on the Windows system WIN, 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 KnapSac operation.

**type**

Command qualifier.

**format**

**/NOTIFY**

**/NOTIFY=username**

**qualifier value**

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

**operations**

SAVE	RESTORE	LIST
		

**related qualifiers**

none

**restrictions**

none

### description

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

### example

```
$ knapsac/image/notify disk$user: win::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 KnapSac 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, KnapSac 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

```
$ knapsac win::myfiles /overlay/select=large.dat *.* (1)
```

(1) The file LARGE.DAT contained in the pacset myfiles.pac on the Windows system WIN will be restored. If a file with the same version number exists, the saved file will be written on top of the existing file.

.....

## /PORT

Specifies the port number to be used for this operation.

### type

Command qualifier.

### format

**/PORT**

### operations

SAVE	RESTORE	LIST
		

### related qualifiers

none

### restrictions

none

### description

The default port number used by KnapSac for TCP/IP communications is 7000. The port number on the client side may be changed using this qualifier.

### example

```
$ knapsac [...] *.* win::dirtree/port=7003 (1)
```

(1) The directory tree starting at the current default directory will be saved to the pacset dirtree.pac which will reside on the Windows system WIN. The port number 7003 will be used for this connection. The port number on the specified server must have been changed to use 7003 also.



## /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 (SYS\$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

```
$ knapsac/image/password disk$user: win::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 KnapSac 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 KnapSac process should execute.

### operations

SAVE	RESTORE
	

### 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

KnapSac resets the priority when it completes execution.

### example

```
$ knapsac/priority=3/image dka0: win::alldisk (1)
```

(1) An image mode backup of the disk will be written to the pacset alldisk.pac on the Windows system WIN at a priority level of 3.



## /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**

### 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

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

### related qualifiers

**[/IMAGE][/INCREMENTAL][/SINCE]**

### restrictions

none

### description

By default, KnapSac does not change dates in a file header. The exception is the backup date field. /RECORD causes KnapSac 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.

### examples

```
$ knapsac/image/record DKA500: win::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.

```
$ knapsac/record=0 win::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.



## /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. KnapSac deletes the existing file and processes the new file with the same version number.

By default, KnapSac will not restore 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

```
$ knapsac win::myfiles /replace *.* (1)
```

(1) The files contained in the pacset myfiles.pac on the Windows system WIN 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.



## /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

RESTORE		LIST
		

### 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

```
$ knapsac win::test /select=*.com *.* (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.

```
$ knapsac/list win::mydir /select=(*.c,*.mar) (2)
```

(2) The files contained in the pacset mydir.pac 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
<b>SAVE</b>

### related qualifiers

**[/BACKUP] [/BEFORE] [/CREATED] [/EXPIRED] [/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:

<b>BACKUP</b>	The backup file time in the file's header
<b>TODAY</b>	The current day at 00:00:00.0 o'clock
<b>TOMORROW</b>	24 hours after 00:00:00.0 o'clock of the current day
<b>YESTERDAY</b>	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

```
$ knapsac/since=today/modified *.for;* win::src (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.

```
$ knapsac/since=backup/modified [000000...] win::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..



## /STATISTICS

Request statistical information regarding the execution process be displayed.

### type

Command qualifier.

### format

**/STATISTICS**

### operations



### related qualifiers

**[/LOG]**

### 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 KnapSac.

### example

```
$ knapsac/statistics sys$system:*.* win::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.

#### KnapSac 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 KnapSac and the operating system used. Specific information about individual files is not given.

### example

```
$ knapsac/list/summary win::* (1)
```

A summary display of each pacset in the backups directory on the Windows system WIN will be produced.

.....

## /SUBDIRS

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

### type

Command qualifier.

### format

**/SUBDIRS**

### qualifier value

none

### operations

DIRECTORY
-----------

### related qualifiers

**[/DIR]**

### restrictions

none

### description

Used with /DIR, this qualifier requests that the contents of all sub-folders also be displayed.

### examples

```
$ knapsac/dir/subdirs win:: (1)
```

(1) The contents of the Windows folder (and all sub-folders) will be displayed.



## /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

```
$ knapsac win::test /truncate *.* (1)
```

(1) The pacset test.pac 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**

### **operations**



### **related qualifiers**

**[/LIST] [/LOG] [/NONE]**

### **restrictions**

none

### **description**

This qualifier requests verification of the pacset.

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 /LOG is also requested, a message indicating successful verification is output as each file is verified.

### **examples**

```
$ knapsac/list/none/verify win::"c:\backups\*" (1)
```

(1) Each pacset in the \backups directory on the Windows system WIN 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 KnapSac.

### type

Command qualifier.

### format

**/VERSION**

### operations



### related qualifiers

none

### restrictions

cannot be used with any other qualifier

### description

Displays the version of KnapSac 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

```
$ knapsac/version (1)
```

(1) The following information will be displayed to SYS\$OUTPUT.

```

KnapSac Version 1.0-0 Evaluation Kit Level D
Copyright 2006 Compact Data Works, Inc.
All rights reserved.
Licensee: EVALUATION KIT
Serial Number: 000000
Node ID: xxxxxxxx
License Type: Evaluation

```



## /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**

```
$ knapsac/zero_data win::"c:\backups\dka0_save" dka0: (1)
```

(1) The pacset dka0\_img.pac on the Windows system WIN will be restored to the DKA0: disk. Any files which have not had their data saved, will be zeroed to the end-of-file marker.

