FCX for VMS User's Guide

Table of Contents

Trademarks

FCX for OpenVMS

Welcome to FCX Version 7 for OpenVMS

New - self-expanding files!

FCX - running on thousands of computers worldwide every day!

Version 6 has new features.

What's New in Version 7?

- Self-expanding files.
- Support for ODS-5 disks.
- Additional file header information preserved.
- Enhanced compression.

Introduction

FCX is a set of powerful and fast file space management products which combine advanced data compression techniques with extensive file handling operations to minimize the amount of disk space required to store files. The FCX product line includes FCX for OpenVMS, Windows, UNIX and Linux. FCX may be used independently or with FCX for Windows, UNIX and Linux providing for VMS file exchange with other systems. FCX may be used to simply save disk space, to archive files online, to speedup file transfers over networks or phone lines, or to exchange files with other systems.

The term "FCX file" is defined to be the output of a compression operation. It contains one or more input files whose data have been compressed. These files are called compressed files. In other words, an FCX file contains one or more compressed files.

An FCX file may be either a sequential or random access FCX file. Sequential FCX files, called transport files, may reside on any media, may contain any number of compressed files and are usually smaller than random access FCX files. Random access FCX files, called discpac files, may only reside on disk. They are truly random access; files may be added to them, removed from them or retrieved from them. Generally speaking, sequential FCX files (transports) are used for saving disk space or for facilitating file transfers; random access (discpacs) files are used for saving disk space, configuration control, and online archival.

Transport operations include:

- Compress disk files into a transport file on disk or tape.
- Compress and append files to an already existing transport on disk.

- Expand the compressed files contained in a transport to their original format.
- List the compressed files contained in a transport.
- Create a self-expanding file for VMS or another supported system.

Discpac operations include:

- Create a discpac file on disk containing any number of input files.
- Remove compressed files from a discpac file.
- Update a discpac file by adding new files to it which may include automatic deletion of old versions of files.
- Refresh the discpac, i.e., search for later versions of files in the discpac and add these later versions to the discpac.
- Extract compressed files from a discpac and create a transport file. (No compression or expansion of data required.)
- Insert files contained in a transport file into a discpac. (No compression or expansion of data required.)

FCX Version 7.0 is compatible with FCX for Windows, Linux, and UNIX, i.e., files compressed on one system may be expanded on the other. The structure of the command syntax is similar as well as use of the operations and most of the qualifiers.

Using this Help

This Help system combines all 3 of the previous FCX manuals: FCX User's Guide, FCX Reference Manual, And FCX Installation and Tuning Guide. Only the FCX messages have been omitted - they are available with the FCX Message Help system.

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

Button Type	Button example
A product or an operation.	
Qualifier	

Other link	Installing FCX

In addition to navigating around with the buttons, browse sequences are provided. The arrows at the top of the Table of Contents will turn red when you enter a browse sequence. You can then use the arrows to view the next and previous pages in the sequence. For example, when you select an operation from the Table of Contents, you have entered a browse sequence for the operations of that product. For each operation, a browse sequence is provided to take you through all the available qualifiers for that operation.

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

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

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

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

Online Help

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

```
$ HELP FCX
```

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

Installation

Installing FCX

Installation Requirements

The installation kit consists of one distribution volume labeled "VAXFCXknm", "AXPFCXknm", or "IA64FCXknm", where k represents the major version number, n represents the minor version number and m represents the revision level.

The installer must have system manager privileges. Approximately 3750 blocks of free disk space are required during the installation procedure; 3000 blocks of disk space are required after the installation is complete. The installation procedure will take about 5 minutes to run depending on the configuration of your system.

The installer should be familiar with the OpenVMS installation procedure VMSINSTAL.COM.

The version of VMS running must be as follows:

- OpenVMS Version 5.5 or higher (VAX)
- OpenVMS Version 6.1 or higher (Alpha)
- OpenVMS Version 8.1 or higher (Itanium)

Installation Options

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

Selecting an Installation Directory

The installation of FCX products requires one or more directories; one directory is required for the FCX common files. A directory may be designated for each product or the products may be installed in the FCX common directory. The logical name FCX_MANAGER is defined to point to the FCX common directory. For historical reasons, the logical name SYS_FCX is also defined to point to the common directory. Examples of files which reside in this directory are:

- A copy of the FCX.CLD file which defines the FCX command.
- The FCX HELP if it is not integrated with VMS HELP.
- Product specific files which do not change from one update to the next. These include the FCX translation library.

Once defined by an installation, future installations (either updates or additional products) will use this same directory for new FCX common files.

Product specific files, e.g., the executable image files, will be installed in the FCX common directory during the initial installation. Updates may be another directory of the installer's choosing. This allows multiple versions of a product to be resident on one system in case of a need to back up a version.

The default for the FCX common directory is to create a system level directory called SYS$SYSROOT:[FCX].

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

```
$ SET COMMAND FCX_MANAGER:FCX
```

This command may also be added to the user's LOGIN.COM command file. Logical names are also set up for the FCX images and the FCX help library if the help was not integrated with the system help files. The image logical names are as follows:

- FCX
- FCXDP
- FCXINS
- FCXEXT

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 calls product specific startup command procedures to define all the FCX logical names. It resides in SYS$MANAGER and should be invoked from the SYSTARTUP.COM file to insure all logical names are defined after the system is restarted. The following command should be added to the SYSTARTUP.COM file:

```
$ @SYS$MANAGER:FCX_STARTUP
```

This command is automatically executed by the installation procedure.

Installing Version 7 Updates

The procedure for installing an update is the same as installing a production kit. When asked during the procedure if you have a license already installed, simply answer YES. You should not need to edit the FCX license files in SYS$MANAGER after the installation.

Conventions used during Installation

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

- User input is underlined.
- All input and output generated as a result of the installation procedure is in bold type.
- The default answer (for those questions that have a 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 2 minutes, depending upon your configuration.

Installation Procedure

The following steps are required to install FCX:

1. Log in to the SYSTEM account.

```
$ Username: SYSTEM

$ Password:
```

2. Invoke VMSINSTAL at the DCL prompt.

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

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

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

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

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

5. Next, you will be asked about the distribution volume. Enter the device from which you are loading the installation kit. This may be the name of a tape drive, CD or disk drive.

```
* Where will the distribution volumes be mounted:
```

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

```
* Products: AXPFCX070, VAXFCX070, or IA64FCX070

* 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 FCX files. If you answer YES, the procedure will create the system level directory **SYS$SYSROOT:[FCX]**. 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 **FCX_MANAGER** will be defined to point to the FCX directory.

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

* Enter the device and directory for FCX installation:
```

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

```
* Do you want FCX commands permanently installed in the DCL tables [YES]?
```

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

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

11. You will now be asked if you want support for multinational character set translation. This capability is provided for converting characters in an FCX file generated on a PC using Thoro'Pac when the language used is not English. See the section on 'Multinational Character Set Translation'. If you are in the United States, you should answer NO.

```
* Do you want support for multinational character set translation [NO]?
```

12. You will now be asked if you already have an FCX license installed. If you are installing an update to a Version 6 product and you already have a license file, answer YES. If you are installing Version 7 for the first time, answer NO even if you already have a license file. The Version 7 license files are different and need to be generated. 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 an FCX license already installed [NO]?

* Enter the FCX evaluation kit key:  XXXXXX
```

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

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

FCX License Files

FCX requires a license file. This is a VMS file with a file name of FCX.LICENSE. A license file is created by the installation procedure and resides in SYS$MANAGER.

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

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

```
$ FCX VERSION
```

The version of FCX 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 contact Innovative Computer Systems, Inc. for instructions on the current license procedures. You may send your current license file to support@fcxfiles.com.

Tuning FCX

Installing FCX as a known image

You may want to install FCX 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 FCX:

```
$ INSTALL

INSTALL> ADD/OPEN/HEADER FCX

INSTALL> ADD/OPEN/HEADER FCXDP

INSTALL> ADD/OPEN/HEADER FCXINS

INSTALL> ADD/OPEN/HEADER FCXEXT

INSTALL> EXIT
```

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

Loading the License File

After installation is complete you may want to load the FCX license file as a logical name. Each license file is a VMS file which resides in SYS$MANAGER and is created as part of the installation. It is verified each time FCX is executed. Loading the license file as a logical name bypasses reading this file each time FCX is executed and provides faster access.

To verify FCX is installed and running properly, simply type the command

```
$ FCX VERSION
```

This command will provide the version of FCX that is executing and the date the evaluation kit will expire (if this is an evaluation kit). To then load the logical name, simply type

```
$ FCX LICENSE/LOAD
```

The above commands are described in the appropriate Operations Section. If there are any problems with the above, remember to log out of the SYSTEM account after installing FCX. Also, remember to execute the SET COMMAND

```
$ SET COMMAND FCX_MANAGER:FCX
```

if you installed FCX with a non-integrated DCL command.

Removing FCX

If you want to remove FCX from your system, the following procedure may be used.

Purging Old Versions

The following steps should be followed to remove FCX after the evaluation period has expired. This procedure will remove all FCX products.

1. All users should remove the SET COMMAND FCX_MANAGER:FCX from their LOGIN.COM files.

2. Log into the System Manager account.

3. Remove the following command line from the SYS$MANAGER:SYSTARTUP_V5.COM command file:

```
@SYS$MANAGER:FCX_STARTUP.COM
```

Delete the command file FCX_STARTUP.COM from the SYS$MANAGER directory.

4. Deassign the logical names:

Care should be taken when deassigning the HLP$LIBRARY logical name. Be sure to do a SHOW LOGICAL/SYSTEM command first to make sure which of the HLP$LIBRARY_1... HLP$LIBRARY_n logical names points to the FCX HELP library.

The commands to deassign the logical names are as follows:

```
$ DEASSIGN/SYSTEM/EXECUTIVE_MODE FCX

$ DEASSIGN/SYSTEM/EXECUTIVE_MODE FCXDP

$ DEASSIGN/SYSTEM/EXECUTIVE_MODE FCXINS

$ DEASSIGN/SYSTEM/EXECUTIVE_MODE FCXEXT

$ DEASSIGN/SYSTEM HLP$LIBRARY

$ DEASSIGN/SYSTEM/EXECUTIVE_MODE FCX_MANAGER

$ DEASSIGN/SYSTEM/EXECUTIVE_MODE SYS_FCX
```

The command to deassign the logical name associated with a loaded license file is as follows:

```
$ DEASSIGN/SYSTEM/USER_MODE FCX_LICENSE
```

5. Delete the contents of the directory SYS$SYSROOT:[FCX] or the directory where FCX was installed

```
$ DELETE SYS$SYSROOT:[FCX]*.*;*
```

6. Delete the directory SYS$SYSROOT:[FCX]

```
$ DELETE SYS$SYSROOT:[000000]FCX.DIR;1
```

Note that there are two such directories:

SYS$SPECIFIC:[000000]FCX.DIR and

SYS$COMMON:[000000]FCX.DIR

7. If the FCX HELP was integrated with the system help library, delete the FCX HELP from the system help library in the SYS$HELP directory with the following command:

```
$ LIBRARY/DELETE=FCX SYS$HELP:HELPLIB.HLB
```

8. 8. To remove the FCX commands from the system DCL tables, execute the following set of commands:

```
$ DCLTAB = F$SEARCH ("SYS$SHARE: DCLTABLES.EXE")

$ SET COMMAND/TABLES='DCLTAB'/OUTPUT= 'DCLTAB'/DELETE=FCX


$ MCR INSTALL

 INSTALL> REPLACE SYS$SHARE:DCLTABLES.EXE

 INSTALL> EXIT
```

The effect of the above set of commands will not take place for users currently logged in. Once these users have logged off and then logged back in, the command will no longer be available to them.

FCX Files

Discpac Files

FCX generates random access FCX files, or discpac files, to provide for true online archival of files. A 'directory' or index of the compressed files is maintained within the discpac file which provides for random access of each compressed file. This allows for faster retrieval of any given file. Multiple versions of files may be maintained within the discpac; these are indicated on a directory listing of the files as primary files and backup files. The total number of versions which may be kept for any given file is specified when the discpac file is created. After that, any UPDATE or REFRESH of the discpac will automatically delete the oldest version of a file when the maximum number is stored. Backup versions of files may be accessed using relative version numbers, i.e., the primary file is relative version 0, the most recent backup file is relative number -1, etc.

Each update of the discpac file (using either UPDATE or REFRESH) may specify a key (identification string) for the set of files which is being compressed. The key may be a date, a version number or some string which uniquely identifies the set of files being added to the discpac. Files may then be accessed (listed, retrieved or removed) using this key.

The discpac operations operate on one discpac file at a time, e.g., a listing may be generated of one discpac file only, as opposed to the transport operations which may LIST or EXPAND many transport files at a time.

Compressed files may be inserted into a discpac or extracted from a discpac. No compression or expansion of data is required. Files contained within a discpac may be extracted for transferring to another site and then inserted into a discpac at the receiving site.

FCX discpac files may also be transferred to a Windows, Linux or UNIX system and expanded using FCX for that system.

Levels of Discpac Files

Every FCX discpac file has a structure level associated with it. This tells the software how to interpret the data contained in the library. FCX Version 7 generates Level X files. The **FCX DISCPAC/VER** command will display the current structure level.

This level is vitally important to discpac files since they are updated in place, i.e., a new copy of the discpac is not generated. You cannot write to a Level S library with Level T software without upgrading the library structure. For most operations this is automatic; you will see a message indicating it is being done.

Transport Files

Sequential access transport files are generated using the FCX COMPRESS operation. Files are compressed and stored sequentially in the transport, hence, these transports may reside on any media, e.g., magnetic tape or multiple floppy disks. Since these files are written sequentially, there is no 'wasted' space in them; they are usually smaller than comparable random access discpac files. However, these files may only be read sequentially and may not be modified.

If you are compressing files to be shipped to another site, either over a network or via removable media, this is the best type of FCX file to generate. Compressed files may be selectively expanded as needed.

Transport files may also be transferred to a Windows, Linux or UNIX system and expanded using FCX for that system.

Levels of Transport Files

Every FCX file has a structure level associated with it. This tells the software how to interpret the data contained in the the FCX file. FCX Version 7 generates Level X. Several previous levels have existed and all of them are still expandable. The **FCX TRANSPORT/VERSION** command will also display the current structure level.

This level is important to some of the new features. If you have older FCX files, a LIST/SUM will tell you the structure of those files.

- you cannot APPEND to a Level J (or prior) file.
- self-expanding files require Level M or later
- ODS-5 files are only supported by Level M and later

Any log files which are generated also contain the structure level.

FCX Command Syntax

FCX Command Syntax

The FCX command is structured in the same manner as all VMS commands. The syntax requires a verb, an operation, zero, one or two parameters (p_disc and p_input) and optional qualifiers.

Format:

(1) $ verb operation p_input [/qualifier-1.../qualifier-n]

(2) $ verb operation p_discpac p_input [/qualifier-1.../qualifier-n]

(3) $ verb operation p_discpac [/qualifier-1.../qualifier-n]

(4) $ verb operation [/qualifier-1.../qualifier-n]

(5) $ verb operation p_input p_output [/qualifier-1.../qualifier-n]

The verb tells DCL which program it is to execute. The verb used to invoke each FCX product is FCX. The operation tells the FCX program which function it is to perform.

Transport Operations

• APPEND	compresses one or more files and appends them to an existing transport file. (Format 5)
• COMPRESS	compresses one or more files into a single transport file. (Format 1)
• EXPAND	expands the contents of one or more transport files. (Format 1)
• LIST	displays the contents of one or more transport files. (Format 1)
• TRANSPORT	provides version and license information. (Format 4)

Discpac Operations

• DISCPAC	provides version and license information. (Format 4)

• CONVERT	converts a discpac from revision date to creation dates or vice versa. (Format 3)
• CREATE	compresses one or more files and creates a discpac file. (Format 2)
• DIRECTORY	displays a directory of an FCX discpac file. (Format 2)
• EXTRACT	extracts compressed files from a discpac and creates a transport. (Format 2)
• INSERT	inserts compressed files contained in a transport into a discpac. (Format 2)
• REFRESH	refreshes files in an existing discpac (newer versions of the files are added to the discpac). (Format 2)
• REMOVE	removes selected files from a discpac. (Format 2)
• RETRIEVE	retrieves and expands files contained in a discpac. (Format 2)
• UPDATE	compresses one or more files and adds them to an existing discpac. (Format 2)

Parameters

The parameter p_discpac (Format 2 and 3) is only required for discpac operations; it identifies the name of a discpac file to be created or operated upon.

The parameter p_input is required for both transport and discpac; it identifies a file or list of files to be operated upon. More than one file specification may be provided, separated by commas. All parts of the second and succeeding file specifications may be omitted. Defaults will be taken from the preceding file specification. This is referred to as sticky defaults. For example, the following two lists of files specify the same set of files:

```
SYS$DISK:[TEST]*.DOC,*.ASM

 SYS$DISK:[TEST]*.DOC,SYS$DISK:[TEST]*.ASM
```

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

Indirect Command Files

The same list of files could have been written to a file and specified as

```
@FILE
```

where FILE.COM contains the list the files. The default file type is .COM. If another file type had been used, it would need to be included on the command line. The command file FILE.COM may also contain qualifiers and other parameters needed for the command line.

Default Input-file-spec

A default for the input-file-spec may also be specified using the logical name FCX_DEFAULT_FILENAME. This is most commonly used to specify a wild card for the file version number and is applicable to transport operations only. For example, if you are using FCX for backup, you may wish to compress all versions of files each time; you would then define the logical name FCX_DEFAULT_FILENAME to be *.*;*. You would then only need specify the device and directory on the COMPRESS command line.

```
$ DEFINE FCX_DEFAULT_FILENAME *.*;*

$ FCX COMPRESS [*...]/OUT=TAPE:ALLDISK
```

These commands would compress the entire disk and create a transport file called ALLDISK.FCX on the device TAPE:.

Full wild card support is provided. This refers to the use of the characters * and %. For a complete description of VMS file specifications, refer to "Guide to OpenVMS File Applications".

Qualifiers

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

```
/qualifier[=value]
```

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

All FCX qualifiers are either global or positional, i.e., they may be placed following the operation or following the parameter (see the Glossary). Global qualifiers affect the entire operation, independent of where they are placed on the command line. Positional qualifiers affect the entire operation if they are placed following the operation. If they are placed following a file specification, they affect the operation on that particular file specification only and override any positional qualifiers placed following the operation.

Note:

> FCX discpac, positional qualifiers placed after the discpac name have no meaning; these qualifiers must be placed after the operation or after a particular file specification.

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

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

Getting Started with FCXtransport

Getting Started with FCX transport files

Using FCX is as simple as copying a file. There are many qualifiers available for selection of input files, redirecting output files and requesting messages during the operation. The qualifiers are discussed in detail in the Qualifiers Sectionl. All messages generated by FCX may be found in the FCX Messages Help.

This section is intended as a tutorial to get you started and illustrate how transport files may be used effectively. Not all qualifiers or variations of qualifiers will be discussed here. Throughout this section, operations and qualifiers are abbreviated in the examples as one might do when actually entering commands.

FCX does not produce any status messages unless it is requested to do so. When first using FCX, it is recommended that you use **/MONITOR** or **/LOG** with all operations. Messages will be displayed indicating which file is being processed, the name of any output files and some statistical information regarding the process. These qualifiers are defaulted ON if you are using an evaluation kit.

TRANSPORT

Verifying FCX is Installed and Available

Once FCX is installed on your system, you may need to issue a SET command to access the FCX command; this is dependent on the type of installation your System Manager selected. The following command will tell you whether FCX 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.

```
$ FCX VERSION
```

If you have any problems with this, see your System Manager.

Installing FCX

Compressing Files

The command to compress a single file is very simple:

```
$ FCX COMPRESS MYFILE.DAT
```

This command will compress the file MYFILE.DAT in your current default directory and create a transport file called MYFILE.FCX also in your current directory. No other output will be generated. If you would like to see statistics indicating how much compression was achieved, you would simply add **/STAT** (STATISTICS) to the command line. All qualifiers and operations may be abbreviated.

If you just want to compress several files and don't need special selection or exclusion qualifiers, the following command will compress your files, let you monitor what is happening, and give you statistics at the end. The transport file will have the name MYFILES.FCX. Only the highest version of each file will be compressed. If all versions are desired, `*.DAT;*` should be used on the command line.

```
$ FCX COMP/STAT/MON *.DAT/OUT=MYFILES
```

If the **/OUT** (OUTPUT) qualifier is not specified then the transport file will have the same name as the first file compressed, but with an extension of "`.FCX`".

/MON (MONITOR) and **/LOG** both give messages about the progress of the compress operation. **/LOG** gives fewer messages and lets you send the messages to a logfile. If you use neither **/MON** nor **/LOG** you won't see any messages until all files have been compressed.

The input file list may consist of any number of files. In the example above, the input file list was simply `*.DAT`. It could easily be expanded to be several sets of files indicated by wild cards and separated by commas.

```
$ FCX COMP/STAT/MON *.COM,*.LIS,*.DAT/OUT=MYFILE
```

Indirect Command Files

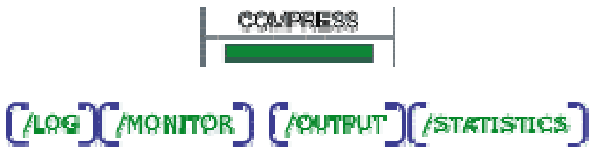
The input file list may also be specified using an indirect command file which contains the list of files. For example, if the file **INPUT.COM** contained the list

```
*.COM,*.LIS,*.DAT
```

then the above command would be specified as

```
$ FCX COMP/STAT/MON/OUT=MYFILE @INPUT
```

This is useful if the list of files is lengthy, awkward to type, or is used many times. Qualifiers and other parameters may also be specified in the command file. Everything on the command line following the indirect command file (in this case @INPUT) is ignored. Additional qualifiers must be placed in the command file. The default file type for the command file is **.COM**.



Changing the Default input-file-spec

There is no default for the file name or type in the input-file-spec. The device and directory will default to the current process defaults. The version number will default to 0, i.e., highest version only.

A default may be supplied using the logical name FCX_DEFAULT_FILENAME. For example, this logical name could be defined to be `*.DAT` as in the second example above, or it may be `*.DAT;*` requesting all versions. If this logical name is set, the second example need not specify an input-file-spec as follows:

```
$ FCX COMP/STAT/MON/OUT=MYFILES
```

The most common use of this logical name would be `*.*;*` where all versions of files are requested as in backup operations.

Expanding Files

To expand MYFILES.FCX simply enter the following command:

```
$ FCX EXPAND MYFILES
```

The compressed files contained in MYFILES.FCX will be expanded in the current default directory. If this directory is the same as the one from which the transport file was created, i.e., files of the same name, type and version number already exist, error messages will be generated. Expanded files retain the same file name, type and version number of the compressed files. To override this condition, one may simply add **/NEW** (NEW_VERSION) to the command line and files will be generated with higher version numbers. Or one may add **/REP** (REPLACE) and the existing files will be deleted and replaced by the expanded ones.

The following command expands all files contained within `MYFILES.FCX` into the directory NEWDIR, gives a message as each file is expanded, and gives statistics at the end.

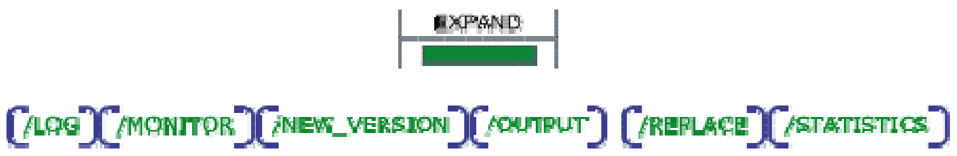
```
$ FCX EXP/STAT/LOG MYFILES/OUT=[NEWDIR]
```

Use **/MON** (/MONITOR) if you want a message as each file is selected for expansion. This is especially useful if you are renaming files as they are expanded or if the original files came from an Windows system. To rename all the files in the transport file, simply add a new extension to the **/OUT** (/OUTPUT) file spec.

```
$ FCX EXP/MON MYFILES/OUT=[NEWDIR]*.NEW
```

In this case, all the files would be expanded into the directory `[NEWDIR]` and all would have the extension `.NEW`. The original file names would be kept intact.

/OUT (OUTPUT) is necessary if you want the expanded files to reside anywhere other than the current default directory.



Listing Compressed Files

Listing the contents of a transport file is accomplished by the following command:

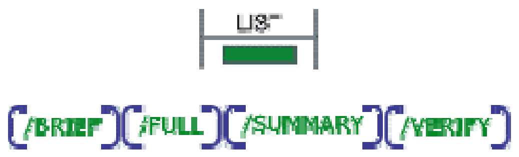
```
$ FCX LIST MYFILES
```

FCX will list to `SYS$OUTPUT` the file specifications of each of the compressed files. This is a brief (**/BRIEF**) list by default. Use **/FULL** if you also want to see file characteristics for each file. **/SUMMARY** may be used to display header information about the transport file only but not each individual file name.

You can see which files are contained within a transport file and verify that the files can be expanded properly using the following command.

```
$ FCX LIST/VER MYFILES
```

/VER (/VERIFY) causes the compressed files to be expanded in memory with CRC checking. If the computed CRC does not match the CRC of the original file, an error message will be output.



Compressing Multiple Files Together

Multiple files can be compressed together. Simply list all the files you want (separated by commas) in the input file specification. You can also use wildcards.

```
$ FCX COMP *.COM,*.DAT/OUT=COMDAT
```

This will compress all the `.COM` and `.DAT` files in the current directory (highest version only). The resultant transport file will be called `COMDAT.FCX`. It will reside in the current directory. If you want the file to be put somewhere else, simply give a directory name such as `/OUT=DUA0:[OTHERDIR]COMDAT`.

The **/OUT** (/OUTPUT) specifies the name of the resultant transport file (default file type is FCX). If you do not specify a name, the transport file will have the same name as the first input file with extension `.FCX`. This may be confusing so it is a good idea to specify the name of the transport file when compressing multiple files together.

You can compress any number of files together. When specifying the list of files, the file specification parts will 'stick', i.e., if you specify a directory name, you need only specify it once if you want all the files to come from the same directory.

```
$ FCX COMP [MYDIR]*.COM,*.DAT
```

This command will compress the `.DAT` files from the directory `[MYDIR]` regardless of the current default.

You may also use a command file which contains the list of files:

```
$ FCX COMP @MYLIST
```

where the file `MYLIST.COM` contains the list of files

```
[MYDIR]*.COM,*.DAT.
```

COMPRESS

/OUTPUT

Indirect Command Files

Directories and Directory Trees

Compressing a Directory Tree

Compressing a directory is a simple matter of using a wildcard input file specification. If **/OUT** (/OUTPUT) is not specified, the transport file will have the same name as the first file compressed but with an extension of `.FCX`. As noted earlier, the use of wild cards must be explicit if all version of files are desired, i.e., `*.*` will find only the highest version of each file. If all versions of all files are desired, `*.*;*` is necessary.

```
$ FCX COM/STAT/LOG *.*/OUT=MYDIR
```

Compressing a directory tree is just as simple, but you need to specify the top or starting directory.

```
$ FCX COM [...]*.*/OUT=MYDIR (start = current dir)

$ FCX COM [MYTOPDIR...]*.* (start = MYTOPDIR)

$ FCX COM [*...]*.*/OUT=ALL (start = root dir)
```

Expanding a Directory Tree

Expanding a directory is also simple. The use of **/OUT** specifies which directory the files are to be expanded into. If it is omitted, the files will be expanded into the current default directory.

```
$ FCX EXP/LOG MYDIR/OUT=[MYNEWDIR]
```

Expanding a directory tree just requires a starting point for the top directory. If **/OUT** is omitted, all the files will go into the current default directory. If the ellipsis (...) is missing, all files will go into the specified directory, i.e., the directory tree will be collapsed.

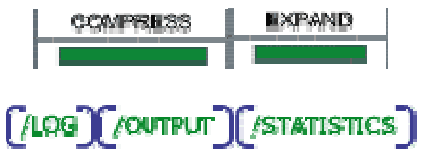
The following command will expand the files and retain the tree structure. If the original input tree contained directories that the tree structure specified by `[MYNEWDIR]` does not, FCX will create the necessary directories to preserve the tree structure.

```
$ FCX EXP/STAT MYDIRTREE/OUT=[MYNEWDIR...]
```

To recreate the directory exactly as it was compressed, use the following command:

```
$ FCX EXP/STAT/LOG MYDIRTREE/OUT=[*...]
```

This command starts the directory from the root directory of the device.



Selecting and Excluding Files

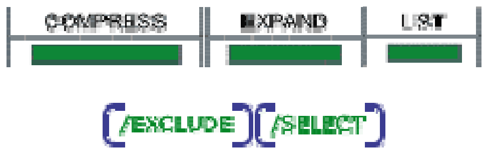
Files can be explicitly excluded from compression, expansion, or listing using the **/EXCLUDE** qualifier as follows.

```
$ FCX COM *.DAT/EXC=(F1.DAT,F2.DAT)/OUT=MOSTDATS

$ FCX EXP/LOG MOSTDATS/EXC=F3.DAT
```

Files can also be explicitly selected for expansion or listing. This is most useful if one only wishes to extract a single file or a small set of files from a transport file.

```
$ FCX EXP MOSTDATS/SEL=(F6.DAT,F7.DAT)

$ FCX LIS/VER MOSTDATS/SEL=F8.DAT
```

When both **/SELECT** and **/EXCLUDE** are used, **/SELECT** is applied first.



Appending Files

The syntax for appending a file is slightly different than for other transport operations. It requires both an input-file-spec and a transport-file-spec (or output-file-spec). The command to append a single file to an already existing transport file:

```
    $ FCX APPEND MYFILE.DAT MYFCX
```

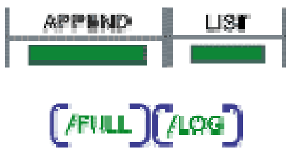
This command will compress the file `MYFILE.DAT` in your current default directory and append it to the transport file called `MYFCX.FCX` also in your current directory. No other output will be generated. Note that transport files Level J or earlier cannot be appended to.

If you want to compress and append several files to another node in your network, the command would look like:

```
    $ FCX APPEND/LOG *.DAT node::[]MYFILES
```

The above command works with VMS nodes, however it is not allowed when the node specified is a Windows node.

The transport file which has new files appended to it looks just like any other transport file, with the exception that the files contained within it may have been compressed using different algorithms. To see this, a **LIST/FULL** is necessary. A brief **LIST** only lists the size and date of each file. For other information about individual files, a **LIST/FULL** is necessary.

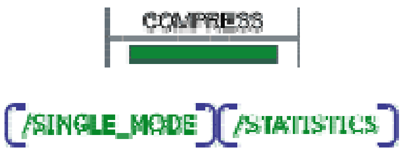


Creating Separate Transport files

It may be desirable, for a number of reasons, to create a separate transport file for each input file. File transfers with unreliable links which tend to break down are one reason. If each transport file is transmitted separately and the link breaks, only the one in progress needs be retransmitted. FCX supports this with the qualifier **/SINGLE_MODE**. Simply attach it to a normal FCX command. Statistics, if requested, will be given separately for each file.

```
    $ FCX COMP/SINGLE_MODE *.DAT/OUT=*.ZZZ
```

This command will result in all the files in the current default directory with extension `.DAT` (highest version only) being compressed. A single transport file will be generated for each file compressed; the transport files will have the same file names as the original files but will have an extension `.ZZZ`.



Getting Started with FCXdiscpac

Getting Started with FCX discpac files

Random access FCX files (discpac files) provide a mechanism to keep several versions of files compressed together in a discpac. A discpac is a 'disk within a disk'. Files within the discpac may be tagged with keys which identify a group of files.

Files within a discpac are either primary files or backup files. Primary files are the latest version of each file; these are the set of files which are listed in the directory by default. Retrieving a file simply by file name will retrieve only the primary file.

Backup files are older versions of primary files. They may have different keys than the primary files. The maximum number of backup files which may reside in the discpac for any given primary file is determined when the discpac is created. Adding files of the same name, after the maximum number of backup files has been reached, results in the oldest backup file being deleted from the discpac. This prevents the discpac from growing too large with unwanted and unnoticed files.

All messages generated by FCX may be found in the FCX Messages Help.

Creating a Discpac File

Creating a discpac file is very simple. Most of the options available may simply be defaulted:

```
$ FCX CREATE MYLIB *.DAT
```

This command will compress the files with file type `.DAT` in your current default directory and create an FCX discpac file called `MYLIB.XLB` also in your current directory. No other output will be generated. If you would like to see statistics indicating how much compression was achieved, you would simply add **/STAT** (STATISTICS) to the command line. All qualifiers and operations may be abbreviated.

If you would like your discpac to have a title when a directory is obtained simply add **/TITLE**=string where string is the set of characters for the title. If you want spaces, you will need to enclose the string in quotes.

The default maximum version count is 10. If you would like a different number add **/MAX_VERSIONS**=n to the command line. This is the number of versions (backup files plus one primary file) allowed for each file.

If you would like this particular set of files (`*.DAT`) to have a special identification string simply add **/KEY**=string to the command line. The following command will create your discpac with a title and a key for each file. Only the highest version of each input file will be compressed and stored in the discpac. Multiple versions of files may reside in the discpac but only one may be added with each operation.

```
$ FCX CREA/TITLE=DAT_FILES MYLIB *.DAT/KEY=VER_1
```

/MON (MONITOR) and **/LOG** both give messages about the progress of the create operation. **/MON** generates more status messages than **/LOG** but **/LOG** lets you send the messages to a logfile. If you use both **/MON** and **/LOG** together all messages will be logged to the logfile. If you use neither **/MON** nor **/LOG** you will not see any messages until all files have been compressed.

Indirect Command Files

The input file list may consist of any number of files. In the example above, the input file list was simply `*.DAT`. It could easily be expanded to be several sets of files indicated by wild cards and separated by commas.

```
$ FCX CREATE/STAT/LOG MYLIB *.COM,*.LIS,*.DAT
```

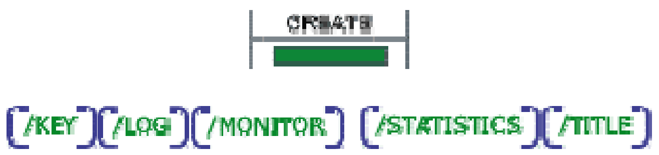
The input file list may also be specified using a command file which contains the list of files. For example, if the file INPUT.COM contained the list

```
*.COM,*.LIS,*.DAT
```

then the above command would be specified as

```
$ FCX CREATE/STAT MYLIB @INPUT
```

This is useful if the list of files is lengthy and awkward to type or is used many times. Qualifiers and other parameters may also be specified in the command file. The default file type for the command file is .COM.



Retrieving a Compressed File

To retrieve and expand one or more files from a discpac file simply enter the following command:

```
$ FCX RETRIEVE MYLIB SAMPLE.DAT
```

The compressed file `SAMPLE.DAT` (primary file only) contained in `MYLIB.XLB` will be expanded in the current default directory.

If this directory is the same as the one in which the original files resided, i.e., files of the same name, type and version number already exist, error messages will be generated. Expanded files retain the same file name, type and version number of the compressed files. To override this condition, one may simply add **/NEW** (NEW_VERSION) to the command line and files will be generated with higher version numbers. Or one may add **/REP** (REPLACE) and the existing files will be replaced by the expanded ones.

The following command retrieves and expands a backup file contained within the discpac `MYLIB` into the directory `NEWDIR` and gives statistics at the end. Backup files may be selected using the key (**/KEY**) associated with them, the VMS version number, or a relative position number.

```
    $ FCX RETR/STAT MYLIB SAMPLE.DAT;-1/OUT=[NEWDIR]
```

If the file to be retrieved had a key of VER_1 the command could be altered as follows and achieve the same result. If the key is specified, it must exactly match that of the file when it was compressed. If you are unsure of the key, use the **DIRECTORY** operation to see exactly what it is.

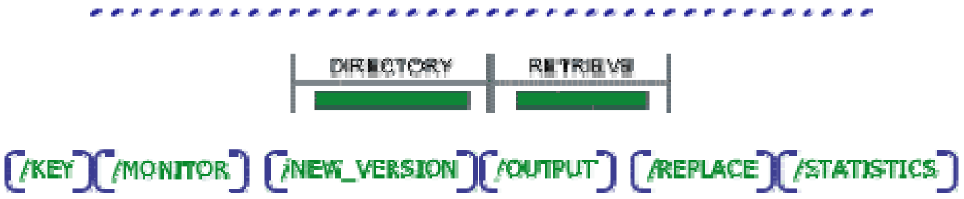
```
    $ FCX RETRIEVE MYLIB SAMPLE.DAT/KEY=VER_1
```

Use **/MON** (/MONITOR) if you want a message as each file is selected for expansion. This is especially useful if you are renaming files as they are expanded or if the original files came from an Windows system. To rename all the files in the discpac file, simply add a new extension to the /**OUT** (/OUTPUT) file spec.

```
    $ FCX RETR/MON MYLIB *.DAT/OUT=[NEWDIR]*.NEW
```

In this case, all the files with file type `.DAT` (primary files only) would be expanded into the directory `[NEWDIR]` and all would have the extension `.NEW`.

/OUT (OUTPUT) is necessary if you want the expanded files to reside anywhere other than the current default directory.



Discpac Directories

Displaying the contents of a discpac file is accomplished by the following command:

```
    $ FCX DIRECTORY MYLIB
```

FCX will display to `SYS$OUTPUT` the file specifications of each of the primary compressed files. This is a brief (**/BRIEF**) list by default. Use **/FULL** if you also want to see file characteristics for each file, the key for each file and the date the file was inserted into the discpac. **/SUMMARY** may be used to display header information but not each individual file name.

If you would like the backup files listed also, simply add **/ALL** to the command line. If you are obtaining a brief directory (this is the default) and would like the file key listed also, simply add **/KEY** to the command line. If you specify **/KEY=string**, only those files whose key matches the specified string will be listed.

Similar to **LIST** with **/SELECT** for transport files, **DIRECTORY** will also process an input-file-spec which specifies which files are to listed in the directory. The following command will list only those compressed files which have file type `.DAT`.

```
    $ FCX DIRECTORY/ALL MYLIB *.DAT
```

You can see which files are contained within a discpac file and verify that the files can be expanded properly using the following command.

```
    $ FCX DIR/VER MYLIB
```

/VER (/VERIFY) causes the compressed files to be expanded in memory with CRC checking. If the computed CRC does not match the CRC of the original file, an error message will be output.

DIRECTORY
/BRIEF
/FULL
/KEY
/SUMMARY
/VERIFY

Updating a Discpac

The **UPDATE** operation provides for adding files to an existing discpac. This may be accomplished simply with the following command:

```
    $ FCX UPDATE MYLIB *.COM
```

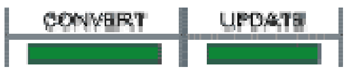
This command will add the files with file type `.COM` from the current default directory to the discpac file `MYLIB.XLB` also in the current default directory. Again, wild cards for version numbers are not allowed. These `.COM` files may be tagged with a key by using the **/KEY=string** qualifier.

```
    $ FCX UPDATE MYLIB *.COM/KEY=VERSION_2
```

If the input-file-spec had also included `.DAT` files, these files would only have been added if they did not already exist in the discpac (FCX checks the revision dates of the files). Use **/MON** (/MONITOR) to see messages indicating when these files already exist.

If the revision date of the file in the current default directory is later than that of the file with the same name in the discpac, the new file is added as a primary file and the existing primary file is moved to be the latest backup file with relative version or position number of ;-1. The backup file with relative position number ;-1 would be moved to relative position number ;-2 and so forth. If the maximum number of versions has been reached, the oldest backup file will be automatically removed from the discpac.

If you would like FCX to change from using revision dates to creation dates or vice versa, use the **CONVERT** operation.



/KEY /MONITOR

Refreshing a Discpac

The **REFRESH** operation is quite similar to the **UPDATE** operation except that it takes as a default input-file-list the contents of the discpac. That is, the only files added to the discpac are later versions of files which already exist in the discpac. The discpac is simply kept up to date or 'refreshed'. The command for doing this is simple:

```
$ FCX REFRESH MYLIB
```

There are very few options for the **REFRESH** command. An input-file-spec may be provided which limits the files in the discpac to be refreshed.

```
$ FCX REFRESH MYLIB *.DAT
```

This command specifies that only the files in the discpac which have file type `.DAT` be refreshed.

As each file is refreshed, the file name is taken from the discpac and compared with the same file name in the original directory. If it is desired to refresh the discpac from the current default directory, use the qualifier **/NODEVDIR**.

```
$ FCX REFRESH/NODEVDIR MYLIB
```

If the revision date of the file in the current default directory is later than that of the file with the same name in the discpac, the new file is added as a primary file and the existing primary file is moved to be the latest backup file with relative version or position number of ;-1. The backup file with relative position number ;-1 would be moved to relative position number ;-2 and so forth. If the maximum number of versions has been reached, the oldest backup file will be automatically removed from the discpac.

Again, you may change from revision dates to creation dates or vice versa using the **CONVERT** operation.

CONVERT REFRESH

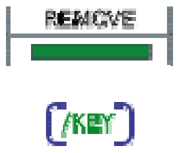
/DEVDIR

Removing Files

Removing files from a discpac file is accomplished using a command such as:

```
$ FCX REMOVE MYLIB SAMPLE.DAT
```

This would result in all versions of the file `SAMPLE.DAT` being removed from the discpac. If you only want one version of the file to be removed, add the VMS version number to the file spec, or use a relative version or position number such as ;-2, or specify a key (**/KEY**).



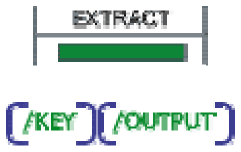
Extracting Compressed Files

If you have a large discpac and wish to transmit some of the files to a remote site, first you will want to extract them in compressed format and create a transport file for the transfer. This is accomplished using a command such as:

```
$ FCX EXTRACT MYLIB *.*/KEY=Latest_Files/OUT=XFER
```

This command will extract all of the files in the discpac `MYLIB.XLB` which have the key string "Latest_Files". None of the files will be expanded. They will instead be put into a new transport file called `XFER.FCX` which is ready for transmit to another site.

Files may also be extracted using the **CREATED** or **MODIFIED** date of each file.



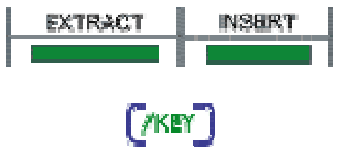
Inserting Files

Complimentary to the **EXTRACT** operation is the **INSERT** operation. Together, these two operations provide a mix and match capability between discpac and transport files. For this example, assume you just received the transport file generated in the **EXTRACT** example. You now wish to insert these files into an already existing discpac on your local system. This can be accomplished with the following command:

```
$ FCX INSERT LOCALLIB XFER/KEY=New_Files
```

The newly received files, contained in the transport file `XFER.FCX`, will be inserted into the discpac `LOCALLIB.XLB`. They will have the new key string "New_Files". As with **EXTRACT**, the files are not expanded and then compressed, thus making these operations very fast.

You will probably want to do a **DIRECTORY** or **DIR/KEY** to see the new files in your discpac. You may now manipulate them as if you had added them with the **UPDATE** operation.


EXTRACT
INSERT
/KEY

FCXtransport Operations

Transport Operaions

FCX generates and manipulates sequential access transport files for transfer to another site, backup, or for online archival.

Format

```
$ FCX  operation  p_input [/qualifier1.../qualifiern]
```

Operations

The APPEND operation provides for adding newly compressed files to an already existing transport file, i.e., input files are compressed and appended to the transport file.

The COMPRESS operation is used to generate a sequential transport file containing one or more files whose data have been compressed. The original files remain intact unless deletion of these files is requested.

The EXPAND operation is used to expand one or more compressed files contained in a transport file. The default mode of EXPAND is to expand all the files; this may be modified using qualifiers.

The LIST operation is used to generate a directory style listing of the contents of one or more transport files. Optionally, it may also be used to verify the data integrity of each compressed file in the transport. This is useful to do after a file transfer if the data is not to be expanded immediately or if there was a suspected transmission error.

Parameters

p_input

The parameter p_input gives the file specification of the file(s) to be operated upon by the specified operation. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied, will default to the current process device and directory. Version numbers, if not supplied, will default to the highest version only.

More than one file specification may be provided, separated by commas. All parts of the second and succeeding file specifications may be omitted. Defaults will be taken from the preceding file specification.

Full wild card support is provided.

Description

Files are compressed and stored sequentially in the transport file, hence, these transport files may reside on any media, e.g., magnetic tape or multiple floppy disks. Any of the algorithms may be used in generating sequential files. Since these files are written sequentially, there is no 'wasted' space in them; they are usually smaller than comparable random access discpac files. However, sequential access files may only be read sequentially and may not be modified. Compressed files may be selectively expanded as needed.

Transport files may also be transferred to a Windows, UNIX or Linux system and expanded using FCX for that system.

APPEND

The APPEND operation is used to compress and append files to an existing transport file. The transport must have been created with Version 4.0 or later of FCX. Except for the syntax, the APPEND operation is virtually identical to the COMPRESS operation. Depending upon the algorithm selected and the actual data in the files, compression may be greater than 90%. The original files remain intact unless deletion of these files is requested. For saving disk space, deletion of the original files would be desired. For transferring files to another site, it would not. Hence, the normal mode of FCX is simply to create a transport leaving the original files intact

Format

```
$ FCX APPEND/qualifier  input-file-spec transport-file-spec
```

Parameters

input-file-spec

The input-file-spec gives the file specification of the file(s) to be compressed. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied, will default to the current process device and directory. Version numbers, if not supplied, will default to the highest version only.

More than one file specification may be provided, separated by commas. Defaults for the second and successive file specifications will be taken from the preceding file specification; any part of the second and successive file specifications may thus be omitted.

Full wild card support is provided.

transport-file-spec

The transport-file-spec gives the file specification of an existing transport to which the compressed files will be appended. There is no default for the file name; the user must supply this. The extension will default to .FCX. The device and directory will default to the current process defaults.

Description

The APPEND operation will compress the input files and append the compressed files to the already existing FCX file specified by the transport-file-spec.

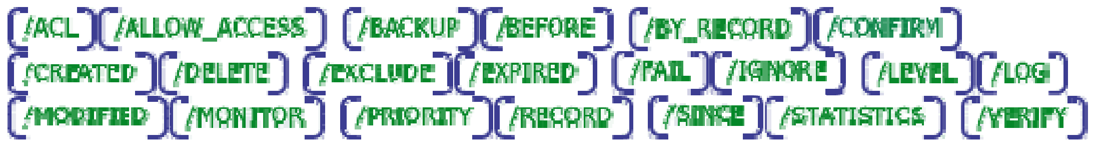
The transport may reside on another node in a DECnet network with the exception of a Windows node.

The transport must be a Level K transport, i.e., it must have been created with Version 4.0 or greater of FCX. The Level of an FCX file may be displayed using the LIST/SUMMARY operation.

Qualifiers

Many of the append qualifiers are used to alter selection of files from the input-file-spec. Files may be selected based on a system time parameter in the file header (see /BEFORE, /SINCE, /BACKUP, /CREATED, /EXPIRED, and /MODIFIED) or file size (/SIZE). Files may be excluded from selection using the /EXCLUDE and /CONFIRM qualifiers. Other qualifiers control execution (LOG, /MONITOR, /PRIORITY, /STATISTICS, /VERIFY).

Qualifier Browse Sequence



COMPRESS

The COMPRESS operation is used to generate a transport containing one or more files whose data have been compressed. Depending upon the algorithm selected and the actual data in the files, compression may be greater than 90%. The original files remain intact unless deletion of these files is requested. For saving disk space, deletion of the original files would be desired. For transferring files to another site, it would not. Hence, the normal mode of FCX is simply to create a transport leaving the original files intact.

The transport may reside on disk, tape, or on a remote node disk.

Format

```
$ FCX  COMPRESS/qualifier  input-file-spec/qualifier
```

Parameters

input-file-spec

The input-file-spec gives the file specification of the file(s) to be compressed. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied, will default to the current process device and directory. Version numbers, if not supplied, will default to the highest version only.

More than one file specification may be provided, separated by commas. Defaults for the second and successive file specifications will be taken from the preceding file specification; any part of the second and successive file specifications may thus be omitted.

Full wild card support is provided.

Description

The compression operation consists of substituting input characters with corresponding compression codes and writing the output transport file. A Cyclic Redundancy Check (CRC) is calculated for each input file and stored in the transport file. This CRC is verified by the EXPAND operation and optionally at the end of the COMPRESS operation (see /VERIFY).

Compression ratios will also vary based on the input data. Files with similar data patterns, e.g., ASCII files, will generate different compression ratios than those with more varied data patterns, e.g., binary files.

Qualifiers

Command qualifiers are used to control the COMPRESS operation.

Many of the COMPRESS qualifiers are used to alter selection of files from the input-file-spec. Files may be selected based on a system time parameter in the file header (/BEFORE, /SINCE, /BACKUP, /CREATED, /EXPIRED, and /MODIFIED). Files may be excluded from selection using the /EXCLUDE and /CONFIRM qualifiers.

Other qualifiers alter the format or destination of the resultant FCX file (/VAR_LENGTH, /OUTPUT) or control the execution process (/LOG, /MONITOR, /PRIORITY, /STATISTICS, /VERIFY).

Password protection may be requested.

Other qualifiers support backup type operations (/RECORD).

Separate FCX files may be generated for each input file using the /SINGLE_MODE qualifier. The /BY_RECORD qualifier supports expansion of selected records from a file.

Self expanding files may be generated.

Qualifier Browse Sequence

[/ACL] [/ALLOW_ACCESS] [/BACKUP] [/BEFORE] [/BLOCK_SIZE] [/BY_RECORD] [/COMMENT]
[/CONFIRM] [/CREATED] [/DELETE] [/EXCLUDE] [/EXPIRED] [/FAIL] [/IGNORE] [/LEVEL]
[/LOG] [/MODIFIED] [/MONITOR] [/OUTPUT] [/PASSWORD] [/PRIORITY]
[/RECORD] [/SELF_EXPAND] [/SINCE] [/SINGLE_MODE] [/STATISTICS] [/VAR_LENGTH]
[/VERIFY]

EXPAND

The EXPAND operation is used to expand one or more compressed files contained in a transport. The default mode of EXPAND is to expand all the files; this may be modified using qualifiers.

The transport is left unchanged.

Format

$ FCX EXPAND/qualifier transport-file-spec/qualifier

Parameters

transport-file-spec

Multiple transport files may be expanded in a single operation. The transport-file-spec gives the file specification of the transport(s) to be expanded. There is no default for the filename; the user must supply this. The file type defaults to .FCX. The device and directory, if not supplied, will default to the current process device and directory. Version numbers, if not supplied, will default to the highest version only.

More than one file specification may be provided, separated by commas. All parts of the second and successive file specifications may be omitted. Defaults will be taken from the preceding file specification.

Description

File information describing each compressed file as well as information describing the compression codes is read from the transport. Each compressed file selected for expansion is then read and reconstructed to match the original input file. As each file is created, a CRC is calculated and compared with the CRC of the original file. If they do not match, an error message is issued.

Expanded files will match the original files both in structure and data content. Many qualifiers are available to override or alter the characteristics of the expanded files.

The EXPAND operation may be used to expand compressed files contained in a transport which was generated on a Windows, Linux or UNIX system. Files on these systems do not have records as do files on a VMS system. Default file information is provided which is used in expanding these files on a VMS system and provides VMS with appropriate record information. The user may override these defaults using the qualifiers /FORMAT and /ATTRIBUTE. The default file information may be displayed prior to expansion using the LIST/FULL operation.

Full wild card support is provided.

Qualifiers

Many of the EXPAND qualifiers are used to alter the file specification or characteristics of the expanded files (/CBT, /CONTIGUOUS, /NEW_VERSION, /OWNER_UIC, /ACL, /REPLACE, /TRUNCATE). /BLOCK_SIZE may be used for expanding files to tape.

Files may be excluded from expansion using the /EXCLUDE, /SELECT and /CONFIRM qualifiers. Other qualifiers alter the destination of the expanded files (/OUTPUT) or control the execution process (/LOG, /MONITOR, /PRIORITY, /STATISTICS. The qualifier /BY_RECORD supports expanding selected records from a file.

Qualifiers are also provided to aid in the expansion of files which originated on a Windows system (/ATTRIBUTE, /FORMAT, /TRANSLATE).

Qualifier Browse Sequence

/ACL /ATTRIBUTE /BLOCK_SIZE /BY_RECORD /CBT /CONFIRM
/CONTIGUOUS /EXCLUDE /FAIL /FORMAT /LOG /MONITOR
/NEW_VERSION /OUTPUT /OWNER_UIC /PRIORITY /RECORD /REPLACE
/SELECT /STATISTICS /TRANSLATE /TRUNCATE

LIST

The LIST operation is used to generate a directory style listing of the contents of one or more transport files. Optionally, it may also be used to verify the data integrity of each compressed file in the transport. This is useful to do after a file transfer if the data is not to be expanded immediately or if there was a suspected transmission error.

The LIST operation provides the display of a transport to the current `SYS$OUTPUT` device. The transport is left unchanged.

Format

$ FCX LIST/qualifier transport-file-spec/qualifier

Parameters

transport-file-spec

The transport-file-spec gives the file specification of the transport(s) to be listed. There is no default for the filename. The file type defaults to `.FCX`. The device and directory, if not supplied by the user, will default to the current process device and directory. Version numbers, if not supplied, will default to the highest version only.

More than one file specification may be provided, separated by commas. All parts of the second and successive file specifications may be omitted. Defaults will be taken from the preceding file specification.

Full wildcard support is provided.

Description

File information describing each compressed file as well as information describing the compression codes is read from the transport file. Each file selected for display is then read and its original file name written to the current `SYS$OUTPUT` device and optionally a log file.

Qualifiers

LIST qualifiers are used to control the format of the directory listing (/BRIEF, /FULL, /SUMMARY). Files may be excluded from the listing using the /EXCLUDE and /SELECT qualifiers. Other qualifiers alter the destination of the listing (/OUTPUT) or control the execution process (/PRIORITY, /STATISTICS). Qualifiers are also provided to verify the integrity of compressed files (/VERIFY). The qualifier, /BY_RECORD, adds the total number of records in a file to the listing if /FULL is also selected.

All LIST qualifiers are global.

Qualifier Browse Sequence

[/BRIEF] [/BY_RECORD] [/DISPLAY] [/EXCLUDE] [/FULL] [/MONITOR] [/OUTPUT] [/PRIORITY]
[/SELECT] [/STATISTICS] [/SUMMARY] [/TRANSLATE] [/VERIFY]

LICENSE

The LICENSE operation provides for loading the FCX license as a logical name.

Format

```
$ FCX  LICENSE
```

Parameters

None

Description

License information from the FCX license file (FCX.LICENSE), located in SYS$MANAGER, is defined as the logical name FCX_LICENSE. This provides for quicker startup of FCX.

FCXdiscpac Operations

Discpac Operations

FCX may be used to generate and manipulate random access FCX files, or discpac files, which provide for online archival of files.

Format

```
$ FCX  operation p_discpac p_input [/qualifier1.../qualifiern]
```

Operations

CONVERT	The CONVERT operation is used to change the default file date and time which is used for updating and refreshing a discpac.
CREATE	The CREATE operation is used to generate a random access discpac file. Only one version of any given file may be included when creating a discpac; wild cards for version numbers are not allowed.
DIRECTORY	The DIRECTORY operation is similar to the LIST operation for transport files. It provides a directory style listing of the contents of a discpac file. Files may be selected for inclusion in the directory listing using the **key** field.
EXTRACT	The EXTRACT operation provides for extracting compressed files from a discpac (with no expansion of data) and creating a transport file to contain the compressed files.
INSERT	The INSERT operation provides for inserting compressed files contained in a transport file into a discpac. No expansion of data is performed.
REFRESH	The REFRESH operation provides an automatic update of the discpac file. FCX looks at the original directory of each file in the discpac to see if there is a later version of the file. If there is, it is added to the discpac as the highest version (primary file). The lowest version of each file will be removed if the maximum number of versions has been reached for that file.
REMOVE	The REMOVE operation is used to remove files from a discpac file. Files may be removed from a discpac by specifying the file name and file type.

The RETRIEVE operation is used to retrieve and expand one or more compressed files contained in a discpac file. This is analogous to the EXPAND operation for transport files.

The UPDATE operation is used to add new files or new versions of files to a discpac file. Each new file is compared with the files in the discpac to see if it is a later version of a compressed file. If it is, it is added as the primary file and the existing files become backup files.

Parameters

p_discpac

The p_discpac parameter gives the file specification of the discpac file to be operated upon. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied, will default to the current process device and directory.

p_input

The p_input parameter gives the file specification of the file(s) to be operated upon. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied, will default to the current process device and directory. Version numbers, if not supplied, will default to the highest version only.

More than one file specification may be provided, separated by commas. All parts of the second and succeeding file specifications may be omitted. Defaults will be taken from the preceding file specification.

Full wild card support is provided.

Description

A 'directory' or index of the compressed files is maintained within the discpac file which provides for random access of each compressed file. This allows for faster retrieval of any given file. Multiple versions of files may be maintained within the discpac; these are indicated on a directory listing of the files as primary files and backup files. The total number of versions which may be kept for any given file is specified when the discpac file is created. When the maximum number for a given file is reached, compressing additional versions of that file will cause automatic deletion of the oldest version of the file. Backup versions of files may be accessed using relative version numbers, i.e., the primary file is relative version 0, the most recent backup file is relative number -1, etc.

Each update of the discpac file (using either the UPDATE or REFRESH operations) may specify a key (identification string) for the set of files which is being compressed. The key may be a date, a version number or some string which uniquely identifies the set of files being added to the discpac. Files may then be accessed (listed, retrieved or removed) using this key.

Discpac files may also be shipped to a Linux, UNIX, or Windows system and expanded using FCX for that systems.

CONVERT

The CONVERT operation is used to change the default file date and time which is used for updating and refreshing a discpac. By default, file revision times are used, i.e., when a discpac is updated, each candidate file for inclusion is compared with those already in the discpac. If the file name matches that of one in the discpac, the file dates are compared. A file is added to the discpac only if its date is later than the one already in the discpac. This operation may be used to change the date processing from revision dates to creation dates or vice versa.

Format

$ FCX CONVERT/qualifier discpac-file-spec

Parameters

discpac-file-spec

The discpac-file-spec gives the file specification of the discpac file to be modified. There is no default for the file name; the user must supply this. The file type will default to `.XLB`. The device and directory, if not supplied by the user, will default to the current process device and directory.

Description

The CONVERT operation is used to change the default file date and time which is used for updating and refreshing a discpac. By default, file revision times are used, i.e., when a discpac is updated, each candidate file for inclusion is compared with those already in the discpac. If the file name matches that of one in the discpac, the file dates are compared. A file is added to the discpac only if its date is later than the one already in the discpac. This operation may be used to change the date processing from revision dates to creation dates or vice versa.

Qualifiers

There is only one CONVERT qualifiers: /DISCPAC_DATE which is used to change the default date used in comparison of files for updating or refreshing a discpac.

Qualifier Browse Sequence

[/DISCPAC_DATE]

CREATE

The CREATE operation is used to generate a random access FCX file or discpac. Only one version of any given file may be included when creating a discpac; wild cards for version numbers are not allowed. This is necessary to maintain the version control that the libraries provide.

Format

$ FCX CREATE discpac-file-spec [input-file-spec] /qualifier

Parameters

discpac-file-spec

The discpac-file-spec gives the file specification of the discpac file to be created. There is no default for the file name; the user must supply this. The file type will default to `.XLB`. The device and directory, if not supplied by the user, will default to the current process device and directory.

input-file-spec

The input-file-spec gives the file specification of the file(s) to be compressed. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied, will default to the current process device and directory. Version numbers will default to the highest version only. Discpacs cannot be created with multiple versions of files.

More than one file specification may be provided, separated by commas. All parts of the second and successive file specifications may be omitted. Defaults will be taken from the preceding file specification.

Full wild card support is provided, except for versions.

Description

An index or directory of the files contained within the discpac file is maintained by FCX within the discpac file itself. This provides for true random access of the compressed files. Each time a file is added to the discpac it is also added to the discpac index. When a file is removed from the discpac, its entry is removed from the index. The data portion of the file is still retained within the discpac until the next time a file is added. The newly added file will use the same file space previously occupied by the removed file.

Inherent with discpac files is the concept of primary and backup files. When a discpac is created, only primary files may be inserted into the discpac, i.e., only one version of the file may be inserted. With subsequent updates to the discpac, newer versions of the primary files may be added. As this occurs, each existing primary file is 'moved' to a backup file. (The file is not actually moved. Only its internal status is updated in the discpac index). Files are moved to backup status on a push down stack basis. This keeps the newest file as the primary file and the oldest file as the lowest version on the stack. The total number of primary and backup files allowed for any given compressed file is specified during discpac creation (See /MAX_VERSIONS). The default is three. Once the limit is reached for any given file, the oldest backup file is removed from the discpac. Backup files may be accessed using relative version or position numbers where ;-1 indicates the most recent backup file and ;-n the oldest. Backup files may also be accessed using the VMS version number.

Files within the discpac may be designated by a key. This is simply an identification string which is used to denote a set of files which were added to the discpac together, either with the CREATE, UPDATE, or REFRESH operations.

Discpacs, since they are accessed randomly, may only reside on a random access device. They may not be created nor updated on magnetic tape. They may however, be transferred to another VMS system or a Windows system. Files may be retrieved on a Windows system in the same manner as sequential FCX files.

Qualifiers

Many of the CREATE qualifiers are used to alter selection of files from the input-file-spec. Files may be selected based on a system time parameter in the file header (see /BEFORE, /SINCE, /BACKUP, /CREATED, /EXPIRED, and /MODIFIED). Files may be excluded from selection using the /EXCLUDE and /CONFIRM qualifiers.

Other qualifiers specify discpac control options (/KEY, /DATE, /MAX_VERSIONS, /TITLE) or control the execution process (/LOG, /PRIORITY, /STATISTICS, /VERIFY). The /BY_RECORD qualifier supports expansion of selected records from a file. Password protection is also available (/PASSWORD).

Qualifier Browse Sequence

[/ACL][/ALLOW_ACCESS] [/BACKUP][/BEFORE] [/BY_RECORD][/CONFIRM]
[/CREATED][/DELETE] [/DISCPAC_DATE][/EXCLUDE] [/EXPIRED][/FAIL] [/IGNORE][/KEY]
[/LEVEL][/LOG] [/MAX_VERSIONS][/MODIFIED] [/MONITOR][/PASSWORD]
[/PRIORITY][/SINCE] [/STATISTICS][/TITLE] [/VERIFY]

DIRECTORY

The DIRECTORY operation is similar to the LIST operation for transport files. It provides a directory listing of the contents of a discpac to the current SYS$OUTPUT device. Files may be selected for listing using the key (/KEY) field. All versions of files may be listed. The default is to list only the primary files (latest version). All versions of each file (primary files and backup files) may be displayed (see /ALL).

The discpac file is not modified.

Format

$ FCX DIRECTORY /qualifier discpac-file-spec [comp-file-spec]

Parameters

discpac-file-spec

The discpac-file-spec gives the file specification of the discpac to be displayed. There is no default for the filename; the user must supply this. The file type defaults to .XLB. The device and directory, if not supplied by the user, will default to the current process device and directory.

comp-file-spec

The comp-file-spec gives the file specification of the compressed file(s) in the discpac which are to be included in the display. The default is to display only the primary files. File names, if supplied, must include the file name and file type. The device and directory will default to all devices and directories included in the discpac. Version numbers will default to the primary file only.

Description

File information describing each compressed file is read from the discpac index. Optionally, files may be verified for data integrity using CRC checking; in this case each file selected is then read and expanded in memory.

Unlike the LIST operation for transport files, files may be selected from the discpac by specifying the desired file specification on the command line. This is similar to using the /SELECT qualifier with the LIST operation. Only one discpac file may be accessed at a time.

More than one comp-file-spec may be provided, separated by commas.

Qualifiers

DIRECTORY qualifiers are used to control the format of the directory listing (/BRIEF, /FULL, /SUMMARY). Files may be selected for or excluded from the directory display using the /ALL, /EXCLUDE and /KEY qualifiers.

Other qualifiers alter the destination of the directory listing (/OUTPUT) or control the execution process (/PRIORITY, /STATISTICS). Qualifiers are also provided to verify the expansion of files (/VERIFY). /BY_RECORD lists the total number of records in the file (if originally compressed using /BY_RECORD).

All DIRECTORY qualifiers are global.

Qualifier Browse Sequence

/ALL /BRIEF /BY_RECORD /DISPLAY /EXCLUDE /FULL
/KEY /MONITOR /OUTPUT /PRIORITY /REBUILD /STATISTICS /SUMMARY
/TRANSLATE /VERIFY

VERSION

VERSION is an operation which provides for reporting the version of FCX.

Format

```
$ FCX  VERSION
```

Parameters

None

Description

The following information will be displayed to SYS$OUTPUT:

```
FCX Version 7.0 Level X
Copyright 1989 - 2005 Innovative Computer Systems, Inc.
All Rights Reserved.
```

The level indicates the internal structure revision level of discpacs and transports to be created with this software. All previous levels are supported for expansion, however, only files with the indicated level will be created.

EXTRACT

The EXTRACT operation provides for extracting compressed files from a discpac file and creating a transport filet. Once the files have been extracted, FCX transport operations apply to the new transport file. The transport may be listed and files expanded from it. The original discpac file remains intact and still contains the compressed files which were extracted. This operation eliminates the need to retrieve files (and expand the data) and then compress them into a transport for file transfers.

Format

```
$ FCX  EXTRACT/qualifier  discpac-file-spec comp-file-spec/OUTPUT=transport-file-spec
```

Parameters

discpac-file-spec

The discpac-file-spec gives the file specification of the discpac file(s) to be accessed. There is no default for the filename; the user must supply this. The file type defaults to `.XLB`. The device and directory, if not supplied by the user, will default to the current process device and directory. Version numbers, if not supplied, will default to the highest version only.

comp-file-spec

The comp-file-spec gives the file specification of the compressed file(s) to be extracted from the discpac. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied by the user, will default to any device and directory. Version numbers will default to the primary file only. Relative version numbers are supported.

More than one file specification may be provided, separated by commas.

transport-file-spec

The /OUTPUT qualifier is required to specify the file name of the transport to be created.

Description

Compressed files are extracted from a discpac in compressed format as transport files. These files may then be treated as other transports and the LIST and EXPAND operations now apply as opposed to the RETRIEVE and DIRECTORY operations. These files may then be transferred to another site or even inserted into another FCX discpac using the INSERT operation.

Qualifiers

Files may be extracted from a discpac in the same manner as they are expanded, i.e., relative version numbers are supported. Files may be extracted by key (/KEY) or by the date of the original files (/BEFORE, /SINCE, /CREATED and /MODIFIED).

The normal selection qualifiers also apply (/CONFIRM and /EXCLUDE). A comment (/COMMENT) may be included in the transport file being created. The /OUTPUT qualifier is required to specify the destination of the transport file. Other qualifiers control the execution process (/LOG, /STATISTICS).

Qualifier Browse Sequence

/BEFORE /COMMENT /CONFIRM /CREATED /EXCLUDE /FAIL /KEY /LOG /MODIFIED /OUTPUT /PRIORITY /SINCE /STATISTICS

INSERT

The INSERT operation is the reverse of the EXTRACT operation, i.e., it inserts compressed files from a transport file into a discpac. Again, no compression or expansion of data is performed. Files may be inserted from transports created on a VMS system or a Windows, Linux, or UNIX system

Format

$ FCX INSERT/qualifier discpac-file-spec transport-file-spec

Parameters

discpac-file-spec

The discpac-file-spec gives the file specification of the FCX discpac file(s) to be accessed. There is no default for the filename. The file type defaults to `.XLB`. The device and directory, if not supplied by the user, will default to the current process device and directory. Version numbers, if not supplied, will default to the highest version only.

transport-file-spec

The transport-file-spec gives the file specification of the transport file whose compressed data are to be inserted in the discpac. There is no default for the filename. The file type defaults to `.FCX`. The device and directory, if not supplied by the user, will default to the current process device and directory.

Description

Compressed files contained within a transport file may be added to a discpac with the INSERT operation. The transport file is not modified. Files are not expanded as they are inserted, only the internal structure is changed. These files may then be expanded from the discpac using the RETRIEVE operation. This operation allows for maintaining a discpac where files are transferred from elsewhere in the form of transport files.

Qualifiers

Files may be selected from the transport file using /CONFIRM, /EXCLUDE, and /SELECT. A key (/KEY) may be used to tag files in the discpac file. Other qualifiers control the execution process (/LOG, /STATISTICS).

Qualifier Browse Sequence

/CONFIRM /EXCLUDE /FAIL /KEY /LOG /PRIORITY /SELECT /STATISTICS

REFRESH

The REFRESH operation is an automatic update of the discpac file. FCX looks at the original directory of each file in the discpac to see if there is a later version of the file. If there is, it is added to the discpac as the highest version (primary file). The lowest version of each file will be removed if the maximum number of versions has been reached for that file. Use of the /NODEVDIR qualifier requests FCX to look at the current directory for later versions of files rather than the original directory.

The p_input parameter specified in the section FCX Command Syntax is optional for this operation; if specified, it gives the file specification of the file(s) in the discpac to be refreshed. The default is to process each file.

Files may also be compressed using the new /BY_RECORD qualifier which allows for expansion of selected records from a file.

Format

$ FCX REFRESH /qualifier discpac-file-spec [comp-file-spec]

Parameters

discpac-file-spec

The discpac-file-spec gives the file specification of the discpac file to be updated. There is no default for the file name; the user must supply this. The file type will default to .XLB. The device and directory, if not supplied by the user, will default to the current process device and directory.

comp-file-spec

The comp-file-spec is optional for this operation; if specified, it gives the file specification of the compressed file(s) in the discpac to be refreshed. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied by the user, will default to the current compressed device and directory.

Version numbers will default to the highest version only; only primary files may be refreshed.

More than one file specification may be provided, separated by commas.

Description

The device and directory of each primary file in the discpac is checked to see if a later version of the file exists. (File dates are compared rather than version numbers (/DATE).) If it does, it is added to the discpac as the new primary file; the existing primary file is pushed down to a backup file. If the maximum number of versions has been reached, the oldest backup file is removed from the discpac. If it is desired to refresh a discpac from a directory other than the original, the qualifier /NODEVDIR may be used.

Qualifiers

REFRESH qualifiers may be used to alter selection of files from the discpac (/KEY and /CONFIRM). Other qualifiers control the execution process (/LOG, /PRIORITY, STATISTICS, /VERIFY).

Qualifier Browse Sequence

[/ACL][/ALLOW_ACCESS][/BY_RECORD] [/CONFIRM][/DELETE] [/DEVDIR][/EXCLUDE]
[/FAIL][/IGNORE] [/KEY][/LEVEL] [/LOG][/MONITOR] [/PRIORITY][/STATISTICS]
[/VERIFY]

REMOVE

The REMOVE operation is used to remove files from a discpac file. Files are removed from a discpac by specifying the file name and file type. By default, all files of the same name are removed, i.e., the primary file as well as all backup files. If a particular version of the file is desired, the VMS version number may be specified or a relative version number. In this case, only the specified file is removed; the remaining backup files and/or the primary file are left in tact. The key (see /KEY) of the file may also be used to select files.

Format

$ FCX REMOVE/qualifier discpac-file-spec comp-file-spec

Parameters

discpac-file-spec

The discpac-file-spec gives the file specification of the discpac file to be updated. There is no default for the file name; the user must supply this. The file type will default to `.XLB`. The device and directory, if not supplied by the user, will default to the current process device and directory.

comp-file-spec

The comp-file-spec gives the file specification of the compressed file(s) to be removed from the discpac. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied by the user, will default to any device and directory. Version numbers will default to the primary file only. Relative version numbers are supported.

More than one file specification may be provided, separated by commas.

Description

By default, all files of the same name are removed, i.e., the primary file as well as all backup files. If a particular version of the file is desired, the VMS version number may be specified or a relative version number. In this case, only the specified file is removed; the remaining backup files and/or the primary file is left in tact. The key (see /KEY) of the file may also be used to select files.

The file space occupied by removed files is returned to a pool of free space in the discpac index. This space is then available for addition of new files. Removal of files does not reduce the size of the discpac.

Qualifiers

REMOVE qualifiers may be used to select files from the discpac-file-spec (/KEY and /EXCLUDE). Optional confirmation of removal is available using the /CONFIRM qualifier. Other qualifiers control the execution process (/LOG, /PRIORITY, /STATISTICS, /VERIFY).

All REMOVE qualifiers are global.

Qualifier Browse Sequence

[/CONFIRM][/EXCLUDE][/FAIL] [/KEY][/LOG] [/PRIORITY][/STATISTICS]

RETRIEVE

The RETRIEVE operation is used to retrieve and expand one or more compressed files contained in a discpac file. This is analogous to the EXPAND operation for transport files.

Files may be retrieved from a discpac by specifying the file name and file type. If a particular version of the file is desired, the VMS version number may be specified or a relative version number may be used. The key (/KEY) of the file may also be used to select files. By default, only the primary file is retrieved.

Selected records of files may now be retrieved using the /BY_RECORD qualifier. These files must have been compressed using the /BY_RECORD qualifier.

The discpac file is not modified.

Format

```
$ FCX  RETRIEVE/qualifier  discpac-file-spec  comp-file-spec
```

Parameters

discpac-file-spec

The discpac-file-spec gives the file specification of the discpac file(s) to be accessed. There is no default for the filename. The file type defaults to `.XLB`. The device and directory, if not supplied by the user, will default to the current process device and directory. Version numbers, if not supplied, will default to the highest version only.

comp-file-spec

The comp-file-spec gives the file specification of the compressed file(s) to be retrieved from the discpac. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied by the user, will default to any device and directory. Version numbers will default to the primary file only. Relative version numbers are supported.

More than one file specification may be provided, separated by commas.

Description

File information describing each compressed file as well as information describing the compression codes is read from the discpac file. Each compressed file selected for expansion is then read and reconstructed to match the original input file. As each file is created, a Cyclic Redundancy Check (CRC) is calculated and compared with the CRC generated for the original file. If they do not match, an error message is issued.

Retrieved files will match the original files both in structure and data content. Many qualifiers are available to override or alter the characteristics of the retrieved files.

Files may be retrieved from a discpac by specifying the file name and file type. If a particular version of the file is desired, the VMS version number may be specified or a relative version number may be used. The key strings (see /KEY) associated with the files may also be used for selection. By default, only the primary file is retrieved.

The RETRIEVE operation may be used to expand compressed files contained in a discpac file which was generated on a Windows, Linux or UNIX system. Files on these systems do not have records as do files on a VMS system. Default file information is provided which is used in expanding these files on a VMS system and providing VMS with appropriate record information. The user may override these defaults using the qualifiers /FORMAT and /ATTRIBUTE. The default file information may be displayed prior to expansion using the DIRECTORY/FULL operation.

Accessing Discpacs on a PC

Qualifiers

Many of the RETRIEVE qualifiers are used to alter the file specification or characteristics of the expanded files (/ACL, /CBT, /CONTIGUOUS, /NEW_VERSION, /OWNER_UIC, /REPLACE, /TRUNCATE). Files may be excluded from expansion using the /EXCLUDE and /CONFIRM qualifiers.

Other qualifiers alter the destination of the expanded files (/OUTPUT) or control the execution process (/LOG, /PRIORITY, /STATISTICS). Qualifiers are also provided to aid in the expansion of files which originated on an Windows system (/ATTRIBUTE, /FORMAT, /TRANSLATE).

All RETRIEVE qualifiers are global.

Qualifier Browse Sequence

/ACL /ATTRIBUTE /BY_RECORD /CBT /CONFIRM /CONTIGUOUS /EXCLUDE /FAIL
/FORMAT /KEY /LOG /MONITOR /NEW_VERSION /OUTPUT
/OWNER_UIC /PRIORITY /REPLACE /STATISTICS /TRANSLATE /TRUNCATE

UPDATE

The UPDATE operation is used to add new files or new versions of files to a discpac file. Each new file is compared with the files in the discpac to see if it is a later version of a compressed file. If it is, it is added as the primary file and the existing files become backup files. If the new file does not exist in the discpac, it is simply added as a primary file. The oldest version of each file will be removed if the maximum number of versions has been reached for that file. Use of the /NODEVDIR qualifier requests FCX to look at the current directory when comparing file names for later versions of files; the device and directory of the files are ignored. Files may also be compressed using the /BY_RECORD qualifier which allows for expansion of selected records from a file.

Format

```
$ FCX  UPDATE/qualifier  discpac-file-spec  input-file-spec
```

Parameters

discpac-file-spec

The discpac-file-spec gives the file specification of the discpac file to be updated. There is no default for the file name; the user must supply this. The file type will default to `.XLB`. The device and directory, if not supplied by the user, will default to the current process device and directory.

input-file-spec

The input-file-spec gives the file specification of the file(s) to be compressed. There is no default for the file name or file type; the user must supply these. The device and directory, if not supplied by the user, will default to the current process device and directory. Version numbers will default to the highest version only. Discpacs cannot be updated with multiple versions of files.

More than one file specification may be provided, separated by commas. All parts of the second and successive file specifications may be omitted. Defaults will be taken from the preceding file specification.

Full wild card support is provided, with the exception of version numbers.

Description

If a file of the same name and type already exists in the discpac, the new file is added as a primary file and the file already within the discpac is 'pushed down' to a backup file. Backup files already within the discpac are pushed down one level and the oldest file removed if the maximum number of versions has been reached.

The device and directory of each file to be added is checked against the device and directory of files already within the discpac. If the full pathname matches, the file is updated as described above (the new file is added as a primary file and the current primary file is pushed down to a backup file). It may be desired to update a discpac from a different device or directory. In this case, only the file name and file type are checked against what exists within the discpac. This is accomplished through the use of the /NODEVDIR qualifier. The device and directory of each file is carried with it in the discpac.

Qualifiers

Many of the UPDATE qualifiers are used to alter selection of files from the input-file-spec. Files may be selected based on a system time parameter in the file header (/BEFORE, /SINCE, /BACKUP, /CREATED, /EXPIRED, and /MODIFIED). Files may be excluded from selection using the /EXCLUDE and /CONFIRM qualifiers.

Other qualifiers specify discpac control options (/KEY) or control the execution process (/LOG, /STATISTICS, /PRIORITY, /VERIFY). Input files may be deleted using /DELETE.

Qualifier Browse Sequence

[/ACL][/ALLOW_ACCESS] [/BACKUP][/BEFORE] [/BY_RECORD][/CONFIRM]
[/CREATED][/DELETE] [/DEVDIR][/EXCLUDE] [/EXPIRED][/FAIL] [/IGNORE][/KEY]
[/LEVEL][/LOG] [/MODIFIED][/MONITOR] [/PRIORITY][/SINCE] [/STATISTICS][/VERIFY]

Advanced Features

Advanced Features

Once you are familiar with the FCX products, you may want to try some new things. FCX supports selective expansion of records from an FCX file, creation and manipulation of VMS libraries on a Windows computer, and can easily be automated through the use of batch files.

Self-Expanding FCX files

Creating a self-expanding FCX file is as simple as adding **/SELF_EXPAND** to the **COMPRESS** command line.

```
$ FCX COMPRESS/SELF *.h/OUT=SELF_H.SFCX
```

will compress all the .h files in the current default directory and create a self-expanding file called H.FCXzz where zz indicates the system you are currently running on. This differentiates self-expanding files from other FCX files.

There are two ways to expand the files: simply run the self-expanding file or define a foreign command.

To run the file, simply type

```
$ RUN SELF_H.FCXzz
```

This will expand the files into the current default directory. You will be prompted for a password if one was used during generation of the file

To define a foreign command, simply type

```
$ SELF_H := $dev:[dir]SELF_H.FCXzz
```

You may now expand the file and redirect the output using **/OUTPUT**. You may list the file using **LIST** or **LIST/VERIFY**. The syntax is slightly different than FCX.

```
$ SELF_H  LIST/VER

$ SELF_H  EXP/OUT=destination
```

You have available all the normal EXPAND qualifiers. For LIST, you have /BRIEF, /SUMMARY, and /VERIFY.

COMPRESS
/SELF_EXPAND

Expanding Selected Records in a File

If you have very large files which you would like to compress but you need access to portions of the files, you may want to try the new feature **/BY_RECORD**. Basically, this allows you to retrieve selected records from a file without expanding the entire file.

FCX is designed to accomplish this task very quickly. Accessing records in a file is similar to accessing files within a discpac, i.e., the entire file need not be read and processed to find the records. In order to accomplish this, files need be compressed using the /BY_RECORD qualifier.

This is available for both transports and discpacs. Files within a discpac may all be compressed using /BY_RECORD but this is not a requirement. For example, if /BY_RECORD is specified on the **CREATE** command, the files added to the discpac at that time will have record information associated with them; if a subsequent **UPDATE** or **REFRESH** does not use the /BY_RECORD qualifier, the newly added files will not have record information associated with them. This is fine.

The same is true with transport files. As another example, a transport file may be created using **COMPRESS** without the /BY_RECORD qualifier and a subsequent APPEND may use it. In either case, using the /BY_RECORD qualifier, FCX will expand only the files which have actually been compressed using /BY_RECORD. You will get an error message for files that have not been compressed using /BY_RECORD.

If you are not sure which files have been compressed with /BY_RECORD, do a **DIR/FULL** or a **LIST/FULL** of the file. For each file which has been compressed using /BY_RECORD, FCX will display the total number of records contained in the file.

The /BY_RECORD qualifier for the compress and list operations is simple; it has no value associated with it.

```
$ FCX COMP/BY_RECORD LARGE.DAT

$ FCX LIST/BY_RECORD/FULL LARGE
```

The above commands will create and list the FCX file using /BY_RECORD.

To expand the above file using /BY_RECORD, you will need to know what records you want to see and where you want them to go. By default, FCX will display them. This is fine if the file is an ASCII file, however, you will not want the contents of a binary file displayed. This could lock up your terminal and/or process. Simply use the /OUT (/OUTPUT) qualifier to specify a file to hold the results of the expansion.

```
    $ FCX EXPAND/BY_RECORD=(START:100,END:200) LARGE
```

This command will display 100 records of the file LARGE.DAT starting at record 100.

You may want the records to go to a file. Simply attach /OUTPUT to the command line and specify a file name. The resulting file will look exactly like the original file except for its size.

You may also specify the number of records to expand as follows:

```
    $ FCX EXPAND/BY_RECORD=(START:100,COUNT:5) LARGE
```

This will expand 5 records of the file starting at record 100. You may specify a START, an END, and/or a COUNT. If START is not specified, the beginning of the file is used. If neither COUNT nor END is specified, the end of the file is used.

It should be noted, since the entire file is not being expanded, no CRC checking is done. CRCs are maintained for entire files only.

Accessing Discpacs on a Windows Computer

It may be the case that you have a large, relatively unused, disk residing on a Windows computer or a CDROM on a Windows computer where you would like to store files, keep them online and be able to retrieve them quickly. If you are using PATHWORKS, this is an easy task. To create an FCX file on the Windows computer, simply specify the node name on the **/OUTPUT** qualifier in the same way as you would a VMS node. FCX does not provide for appending to an FCX file on a Windows computer.

FCX does support complete maintenance of VMS discpacs on Windows nodes. You can create discpacs, update and refresh them as if the node were another VMS node. The discpacs, however, will grow somewhat larger with every update, in addition to the files being added. If the device is a WORM drive, this is of little consequence.

Files created in this manner on a Windows computer may be read using FCX Version 7 for Windows. It is important that the FCX level of the file be less than or equal to the level of the software being used to read the FCX file. Software levels are displayed using the **/VERSION** qualifier on the product operation. The level of an FCX file is displayed using the LIST or DIRECTORY operations.

Invoking FCX from within Batch Files

If you are compressing a file or set of files containing tens or hundreds of thousands of blocks, you may want to invoke FCX from within a batch command file. If FCX was installed on your system using the non-integrated approach remember to issue the following command before invoking FCX within your batch file.

```
$ SET COMMAND FCX_MANAGER:FCX.CLD
```

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

Following is a sample command file which invokes FCX to expand a program source file from an FCX file, and checks the status of the EXPAND operation before proceeding.

```
$!       REBUILD.COM - expands PROGRAM_1 from an FCX file

$

$ SET NOON ; disable error checking

$ FCX EXPAND PROGRAM_1.FCX

$ IF $STATUS THEN CC PROGRAM_1 ; if OK then compile

$ IF $STATUS THEN LINK PROGRAM_1 ; if compile OK then link

$ IF $STATUS THEN RUN PROGRAM_1 ; if link OK then run

$ SET ON  ; restore error checking
```

Selecting the Compression Algorithm

FCX Version 7 has two algorithms which offer different degrees of compression effectiveness and speed. They are specified using the **/LEVEL** qualifier.

Each algorithm generates variable length bit codes based on recurring sequences of byte values in the input file data. Each generates good compression with savings as high as 90%. The best compression is usually achieved using **/LEVEL=1**. It is not as fast as **/LEVEL=0** but may generate compression ratios from 2% to 8% higher.

/LEVEL=0 is the default and need not be specified on the command line. If you desire extra compression, you may specify **/LEVEL=1**. On files from 300,000 blocks up to 2,000,000 blocks or higher, you may save from 10,000 to 20,000 blocks depending on the data. For small files, the results will be less significant.

For sets of files, especially ASCII files, consider using **/GROUP**. **/GROUP** invokes **/LEVEL=1** and can achieve even greater compression. The set of files is considered as one big file. Compression is faster and usually 2-5% better. It will take longer to retrieve any one given file, but expanding the entire set will be the same as without **/GROUP**.

The compression results can be determined without generating a compressed file by directing the output to the null device.

[/GROUP][/LEVEL]

Tape to Tape Copy

FCX can compress directly from tape to disk. Files in the FCX file can then be expanded to disk or expanded back to a different tape providing a byte for byte copy of the original tape.

Compression from tape/expansion to tape follows one of two scenarios depending on whether the tape was mounted foreign. If you have no idea how the tape was written or you desire to make an exact copy of it, it should be mounted foreign. If the tape was generated using ANSI labels, it may be mounted with label processing (i.e., not foreign). The following examples will describe the FCX processing.

Example 1:

The tape is an ANSI labeled tape, e.g., it is a VMS distribution kit which was written with BACKUP. The tape may be mounted with label processing; VMS will process the VOL and HDR records. FCX is invoked in the same manner as disk files.

```
$ MOUNT/OVER=ID MUA0:

$ FCX COMP/OUT=TAPE MUA0:*.*
```

FCX will compress all the files on the tape and create a transport file called `TAPE.FCX` in the current default directory. These files may be expanded to disk or to another tape. If the tape is to be an exact copy, it must be mounted with the appropriate block size. In this case, the block size should be 8192. This may be determined by looking at the block size of the files in the transport file using

```
$ FCX LIST/FULL TAPE
```

Although standard ANSI labeled tapes do not use a block size of 8192, in this example BACKUP was used to write the distribution kits to tape in this manner. To recreate the tape

```
$ INIT MUA0: label

$ MOUNT/BLOCK=8192/OVER=ID MUA0:

$ FCX EXP/OUT=MUA0: TAPE
```

In the example above, you should now be able to use BACKUP to list the contents of the tape. The tape needs to be mounted foreign in order to use BACKUP.

Example 2:

Assume the tape was written as above but knowledge about the tape is unknown. The tape is mounted foreign and FCX is used to compress the contents of the tape.

```
$ MOUNT/FOREIGN MUA0:

$ FCX COMP/OUT=TAPE MUA0:
```

FCX will compress all the files on the tape up to the logical end-of-tape. This includes the header and trailer records which will be present on this tape. In this particular example, the transport file will contain 3 files called `FILE1.DAT`, `FILE2.DAT` and `FILE3.DAT`. `FILE1.DAT` will contain the header records, `FILE3.DAT` will contain the trailer records and `FILE2.DAT` will contain the BACKUP saveset. If additional files were present, the file name would continue as `FILEn.DAT`. A LIST/FULL of the transport file will show that `FILE1.DAT` and `FILE3.DAT` contain small files of 80 byte records. `FILE2.DAT` will contain a large file of 8192 byte records. If `FILE2.DAT` were expanded to disk, you could use BACKUP to list the contents of the saveset.

To recreate the tape, mount the new tape as foreign and specify the block size as above.

```
$ MOUNT/FOREIGN/BLOCK=8192 MUA0:

$ FCX EXP/OUT=MUA0: TAPE
```

This tape will be an image copy of the original tape complete with the header and trailer records.

If the original tape was not a labeled tape, the transport file would contain only data files. This should be apparent from the number of files in the transport file and the size of the blocks. Mount the output tape with a block size that corresponds to the block size of the files in the transport file. Expanding to tape will then create an image copy of the original tape.

Saving Disk Space

Saving Disk Space

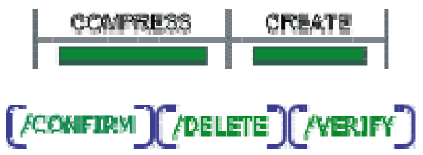
A common problem on many computer systems is a shortage of disk space. Most users have files which are seldom used, but which cannot be deleted because they may be needed at some future time. These files could be backed up to tape but restoring them is a time-consuming process. FCX is used to reduce the online storage requirements for files while retaining online access to the files.

In order to save disk space, the original files need to be deleted after they have been compressed. FCX will do this only if the **/DELETE** qualifier is specified on the **COMPRESS** (or **CREATE)** operation. If the COMPRESS (or CREATE) operation detects any errors, no files will be deleted. However, it is still a good idea to verify the integrity of the FCX file before deleting any files. FCX will do this if the **/VERIFY** qualifier is specified. Compressed files are expanded in memory with CRC checking to verify they can be expanded at a future time. If /VERIFY is specified, all files are verified before any are deleted. If any errors occur, FCX will not delete any files. It is then up to you to manually delete any files.

The following command will compress, verify, and delete the contents of an entire directory and you will be able to monitor the progress. All that will remain at completion of the operation is the resultant transport file (MYDIR.FCX).

```
$ FCX COMP/MON/VER/DEL/OUT=MYDIR.FCX *.*;*
```

If you only want selected files to be deleted during this operation, the **/CONFIRM** qualifier may be used. A prompt is issued before each file is compressed and again before each file is deleted. Any negative response results in the file being skipped. If you respond negatively to a file being compressed, the prompt for that file to be deleted will be skipped, i.e., only files which have successfully been compressed may be deleted.



Compressing Files which need Frequent Access

If you need to access compressed files frequently, such as CAD files which you may want to access daily, consider compressing files individually or in small groups. If there are several related files, such as text and graphics files, which together comprise a picture or drawing or image, compress the related files together into one FCX file.

If you will be updating the files, consider compressing them into a discpac using FCXdiscpac. Retrieval of files from a discpac is faster than from a transport file. After the files have been modified, the discpac may be updated using either the **REFRESH** or the **UPDATE** operation. If disk space is a premium, set **/MAX_VERSIONS=1** when creating the discpac; all older versions of files (i.e., all but the primary file) will be purged as the discpac is updated.

/MAX_VERSIONS

File Transfer

File Transfer

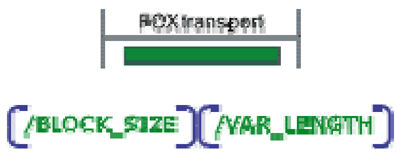
File transfer operations may be speeded up considerably if the files at the sending node are first compressed and then only the smaller transport file is transferred. Multiple files can be transferred in a single operation by packaging the files together into a transport file. FCX is then used to reconstruct the original files at the destination by expanding them from the transport file. The amount of data transferred is reduced and the number of connections required is reduced since only one file is being transferred rather than many. Qualifiers have been established to facilitate the transfer of files, e.g., **/BLOCK_SIZE** and **/VARIABLE_LENGTH**. These may be varied to maximize the performance of a particular file transfer protocol.

Some file transfer protocols do not provide for the transfer of files with specific file structure requirements; i.e., files with fixed length records or indexed files. FCX may be used to compress these files before transferring them. The files are expanded at the receiving node and the file structure, as well as the data content, will exactly match that of the original files.

Many file transfer programs provide for the transfer of multiple files in one operation by providing wild card support. However, a separate data connection is opened and closed for each file. By compressing these files first and then sending only the transport file, only one data connection is opened and closed, resulting in reduced overhead on the network.

By using FCX to compress a directory or a directory tree, one file transfer operation is all that is needed to send the directory or directory tree to the receiving node. FCX is then used to recreate the contents and structure of the directory and/or directory tree at the receiving node.

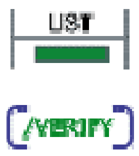
By default, transport files contain fixed length records. As noted above, some file transfer protocols do not support this. **/VARIABLE_LENGTH** may be used to change this.



Verifying Data Integrity after a File Transfer

Use LIST/VERIFY to determine if your transport files arrived intact and the files within them can be expanded.

```
$ FCX LIST/VER MYFILES
```



Compressing on One Computer and Expanding on Another

FCX automatically preserves all file characteristics with the exception of ACLs. These may not have any meaning on another system but if it is desired to preserve them, use the **/ACL** qualifier. This qualifier must also be used when expanding files. If ACLs are preserved when files are compressed but not desired on the target system, simply do not use **/ACL** when expanding the files.

All other file characteristics will be carried over, including the file UIC. FCX will attempt to expand files using the UIC of the original files. If the user initiating the expansion of the files does not have sufficient privilege to access the original UIC, FCX will expand the files using the UIC of the current user. The qualifier **/OWNER_UIC** may be used to specify the UIC of the expanded files.

If the directory in which the expanded files are to be created contains files with the same name and file type as the compressed files, the file access protection information will be taken from the existing files rather than the default for the directory or process.

The qualifiers **/CBT**, **/CONTIGUOUS** and **/TRUNCATED** may be used when expanding to affect characteristics of the expanded files.

/ACL /CBT /CONTIGUOUS /OWNER_UIC /TRUNCATE

Compressing on VMS and Expanding on Windows, Linux, or UNIX

Files compressed on a VMS computer may be expanded on a Linux, Windows, or UNIX computer using FCX Version 7 for that system.

When VMS ASCII files are expanded on a Windows computer, an EOF (End of File, specifically a CTRL-Z character) is appended. Use the FCX **/NOEOF** qualifier when expanding to prevent the EOF from being added.

VMS files which have Carriage Return Carriage Control, Stream, Stream-LF, Stream-CR, Print File, or FTN (FORTRAN) formats will be expanded on the Linux, UNIX, or Windows system with proper carriage return and line feed combinations.

If you want to expand a VMS file which has no counterpart, such as indexed or relative files, you can use **/RAW_DATA** when expanding on the non-VMS computer. The expanded file will be a byte-for-byte image of the VMS file, but will require custom application software on that system in order to make sense of it.

Compressing on Windows, Linux, or UNIX and Expanding on VMS

When a file is compressed on a Windows, Linux, or UNIX system, FCX determines whether the file contains only ASCII byte values (0-127 decimal) or also contains non-ASCII byte values. FCX also determines whether the file contains carriage returns and linefeeds. This information is used by FCX for VMS when expanding these files on VMS to determine the default file format and record attributes.

Use **LIST/FULL** on VMS to see what defaults would be used if the files were to be expanded.

The default file format and record characteristics can be overridden when the files are expanded by using the **/CONFIRM=OVERRIDE**, **/ATTRIBUTE**, and/or **/FORMAT** qualifiers.

[/ATTRIBUTE] [/CONFIRM] [/FORMAT]

Using DECNet

Files may be compressed and sent across a network in one operation if DECnet is running on both the sending and receiving system. You simply need specify **/OUT=node::** on the COMPRESS command line. Of course a device, directory and file name may also be specified.

```
$ FCX COMPRESS/OUT=othernode:: *.COM
```

will result in all the `.COM` files (highest version only) in the current default directory being compressed. The resultant transport file will reside on the DECnet node `othernode::`. It will reside in the current default device and directory on the remote node. The file name will match that of the first file compressed, the file type will be `.FCX`. No transport file will be created on the local node.

This operation is similar to copying a file except that the files will all be compressed together and only compressed blocks shipped across the network; the resultant destination is the same.

Using FCX in this manner with DECnet is faster than compressing files and then copying the resultant transport file. The file transfer may be initiated while the compression operation is proceeding.

If you want to expand a transport file which resides on a remote node, the command is similar to

```
$ FCX EXPAND othernode::device:[directory]TEST
```

The transport file `TEST.FCX` which resides on the remote node `othernode::` will be retrieved across the network and expanded into the current local default directory.

No expanded files will be created on the remote node.

[/OUTPUT]

FCX for Backup

FCX for Backup

FCX provides for backing up files disk to disk or disk to tape. If an entire disk is being compressed for offline storage, use of the qualifier **/IGNORE=INTERLOCK** may be desirable. This provides for inclusion of files which are in use by another process or user. A warning message will be generated indicating the file being compressed has an access conflict, i.e., the data being compressed may be altered during compression.

Files which are marked NOBACKUP will, by default, be included but the data portion of the file will not be compressed. On expansion, the file will be created, blocks allocated for it, but the data will have been lost. This is consistent with the way in which VMS BACKUP works. If it is desired to have the data compressed, simply use the **/IGNORE=NOBACKUP** qualifier.

Also, it may be desirable to set the logical name **FCX_DEFAULT_FILENAME** to include all versions of files, e.g., it may be set to `*.*;*`.

/IGNORE
FCX_DEFAULT_FILENAME

Compressing to Magnetic Tape

FCX can compress files directly to tape and expand directly from tape. Simply mount the tape before invoking FCX, and initialize the tape if necessary. The **/OUTPUT** qualifier is required if compressing to tape in order to specify the tape device.

FCX uses standard tape handling software (ANSI standard format). Refer to "Guide to OpenVMS Disk and Magnetic Tape Operations" for details of this format. In order to use tape as a medium for transport files, it is first necessary to initialize and mount the tape. Block sizes for tape may be varied using the `MOUNT` command; the recommended block size is 32256. FCX will execute faster with this higher block size. FCX will set appropriate internal block sizes based on the setting of the tape when it is mounted. Some tape drives cannot support a block size this large. Refer to the appropriate hardware manual for the maximum block size your tape drive can support.

Sample command lines to initiate a **COMPRESS** operation to tape follow.

```
$ ALLOCATE MSA0:

$ INITIALIZE MSA0: FCXTAP

$ MOUNT/BLOCK_SIZE=32256 MSA0: FCXTAP

$ FCX COMPRESS/VERIFY/OUT=MSA0: *.*

$ DISMOUNT MSA0:
```

The command lines necessary to expand a transport file which resides on tape are nearly the same as above, however, the tape should be write-locked and the `INITIALIZE` command not executed. Sample command lines to initiate an **EXPAND** operation from tape follow.

```
$ ALLOCATE MSA0:

$ MOUNT MSA0: FCXTAP

$ FCX EXPAND MSA0:[]FCXFILE

$ DISMOUNT MSA0:
```

COMPRESS | EXPAND

[/OUTPUT]

Monitoring Performance

Using Statistics to Tune Performance Execution

The CPU and elapsed times given in the FCX statistics (**/STATISTICS**) are measured from the time FCX begins processing the input file specification until processing is completed. The CPU time indicates the actual amount of time FCX took to process the files. The elapsed time is the wall clock time which elapsed from when FCX began processing until processing was completed.

Since FCX overlaps I/O with the compression calculations, the two times will be very close to each other if FCX is the only process executing on the machine. If there is a large discrepancy between the times, then other processes are also executing and will affect how quickly FCX completes its processing. If the CPU time indicates 30 seconds to compress files, but the elapsed time is 5 minutes, then either your system is heavily loaded or other processes are running at higher priorities than the process which is executing FCX.

Page Faults of a few hundred to a few thousand pages are considered normal. However, if the statistics indicate excessive page faults you probably have insufficient working set quota which will result in degraded performance. If FCX is taking an excessively long time to execute, this is the parameter to look at. You can estimate the amount of working set quota you need by looking at the Virtual Memory Used statistic, which is given in pages (512 bytes). Increase your working set quota or extent to accommodate the required virtual memory. Suggested minimum working set quotas are 1500 pages.

[/STATISTICS]

Using CTRL-T to Monitor Progress

During compression (**COMPRESS**, **REFRESH**, and **UPDATE**) and expansion (**EXPAND**, **RETRIEVE**) operations, FCX 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 compressed or expanded and the block number of the file that FCX is currently processing. This is most useful if you are compressing very large files and would like to see 'progress reports' as the operation proceeds.

Setting the Process Priority

Compression of large files may consume substantial system resources such as CPU time and memory. If you are running FCX interactively, or even from a batch job, you may wish to lower the priority of the process executing FCX. This is accomplished easily using the **/PRIORITY** qualifier. The following example will compress a file using the large algorithm and set the process priority to 3.

```
$ FCX COMP/PRI=3 large.dat
```

The ALTPRI privilege is required for the priority to be changed. If you do not have this privilege, the /PRIORITY qualifier will have no effect. At the end of the compression, the priority is changed back to what it was before the /PRIORITY qualifier took effect.

/PRIORITY

Error Processing

Error Processing

All error messages, as well as informational messages, generated by FCX are described in the FCX Messages Help system. FCX will continue to process files unless a fatal error has occurred. If **/FAIL** has been used, FCX will stop processing on non-fatal errors. The action to be taken by the user will depend on the actual error.

Use of `<CTRL-C>` and `<CTRL-Y>` to abort an FCX operation will have no significant effects if the operation in progress is on a transport file. Processing will simply terminate.

Discpac Operations

During a discpac operation, aborting the operation via a `<CTRL-C>` or `<CTRL-Y>` will have no significant effects on the discpac. If the operation in progress is an update type (**REFRESH**, **UPDATE** or **REMOVE**) the discpac will contain the files added to it (or removed) as of the instance of the key press. Any file in progress will not be affected, i.e., it will not have been added (or removed).

During creation of a discpac, use of `<CTRL-C>` or `<CTRL-Y>` will cause the operation to abort, but the discpac file will have been created and is not deleted. The discpac may contain files but they will be inaccessible. A **DIRECTORY** of the discpac will indicate that the discpac is empty. In this case, it is best to delete the discpac file and start over.

If a discpac operation was accidentally aborted (system crash etc.), it must be rebuilt before any write operation can be done. A journal file is kept and normally discarded at the end of the discpac operation. After a system crash, the journal file should still be there. The file will have an extension of `.JOU` and will reside in your login directory. DO NOT DELETE THE JOURNAL FILE. It will be used to recover the discpac. This recovery is done automatically with all write operations except for **REMOVE**. It may be recovered manually by using the **DIR/REBUILD** command. The journal file will then be deleted by FCX. You may wish to do a **DIRECTORY/FULL/ALL** to see exactly the state of the discpac. If you are concerned about data integrity, add **/VERIFY** also.

Error Logs

Error messages, as well as other messages, are logged to a log file if there was one specified on the command line. In addition, error messages are also displayed to `SYS$OUTPUT`. There are many times, however, when an error message may be overlooked. FCX will create an error log file, if there is not a log file, and write error messages to this error log file. This special error log file is only created when an error occurs. It will reside in the current default directory and have the name `FCXERROR.LOG`. At the end of an operation, a message is sent to `SYS$OUTPUT` indicating if any errors did occur and what log file the user may find them in. The error log files are transient and should be purged periodically.

[/FAIL]

Qualifiers

Qualifiers Reference

Each qualifier is presented with a detailed description of its function; applicable operations, its format and whether it is a positional or global qualifier.

Global qualifiers affect the entire operation and may be placed anywhere on the command line. /STATISTICS is an example of a global qualifier.

Positional qualifiers affect the particular file specification they are following; if positioned after the operation, they affect all file specifications. /BEFORE is an example of a positional qualifier.

Most qualifiers may also take a negated form, i.e., the qualifier name is preceded with NO, e.g., /NOLOG. In this form, the qualifier has no value and usually serves to negate any previous qualifier of the same name.

Note:

FCX discpac positional qualifiers placed after the discpac name have no meaning; these qualifiers must be placed after the operation or after a particular file specification.

Qualifier Browse Sequence

/ACL /ALL /ALLOW_ACCESS /ATTRIBUTE
/BACKUP /BEFORE /BLOCK_SIZE /BRIEF /BY_RECORD
/CBT /COMMENT /CONFIRM /CONTIGUOUS /CREATED
/DELETE /DEVDIR /DISCPAC_DATE /DISPLAY
/EXCLUDE /EXPIRED
/FAIL /FORMAT /FULL
/IGNORE
/KEY
/LEVEL /LOG
/MAX_VERSIONS /MODIFIED /MONITOR
/NEW_VERSION
/OUTPUT /OWNER_UIC /PASSWORD /PRIORITY
/REBUILD /RECORD /REPLACE
/SELECT /SELF_EXPAND /SINCE /SINGLE_MODE /STATISTICS /SUMMARY
/TITLE /TRANSLATE /TRUNCATE
/VAR_LENGTH /VERIFY

/ACL

Request that ACLs be preserved during compression and during expansion.

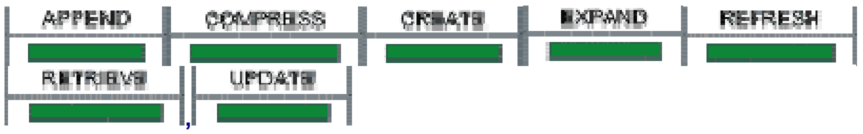
type

Positional qualifier.

format

/ACL
/NOACL (default)

operations



related qualifiers

/OWNER_UIC

description

Information on each input file is stored in the FCX file in addition to the actual file data. By default, enough information is kept to allow the file to be expanded and retain its original structure. Since ACLs may be quite lengthy, they are not normally preserved. Use of this qualifier on COMPRESS or CREATE operations causes ACLs to be saved in the FCX file. Use of this qualifier on EXPAND or RETRIEVE operations causes saved ACLs to be preserved in expanded files.

note

If the directory in which the expanded files are to be created contains files with the same name and file type as the compressed files, the file access protection information will be taken from the existing files rather than the default for the directory or process.

If the expanded output file is to have a UIC different from the user executing the EXPAND operation, that user must have sufficient privilege. See /OWNER_UIC.

examples

```
$ FCX COMPRESS MYDIR:*.*/ACL/OUT=MYDIR (1)
```

(1) All files in the directory pointed to by the logical name `MYDIR` will be compressed. ACLs for each file will be included in the resultant transport file, `MYFILE.FCX`, which will reside in the current default directory. The relative version number will be 0 and the file type will be `.FCX`.

```
$ FCX EXPAND/ACL MYDIR (2)
```

(2) All files in the transport file `MYDIR.FCX` will be expanded. The original file ACL information will be propagated to the newly expanded files. The expanded files will reside in the current default directory.

/ALL

Requests all versions of files contained within a discpac file be displayed.

type

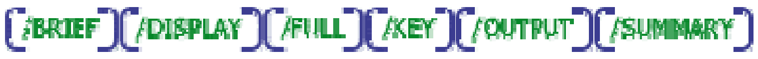
Global qualifier.

format

/ALL

operations

related qualifiers



description

By default, the directory of a discpac file only lists the highest version (primary file) of each file contained within the discpac. /ALL provides for the display of every file within the discpac. The contents of the display will vary with the other qualifiers selected. The default is a brief (/BRIEF) display.

example

$ FCX DIRECTORY/ALL IGSLIB (1)

(1) Sample output, as follows, will be displayed to SYS$OUTPUT:

```
FCX Library File Information
FCX file: DUA0:[DEMO]IGSLIB.XLB;1
Creation date: 21-AUG-1993 20:05:21.22
Date of last update: 21-AUG-1993 20:13:49.08
FCX Version: FCX Version 4.0-0
Operating system: VMS Version V5.5
Algorithm: Fixed Code Dictionary
Max Versions: 3
Number of primary files: 3
Number of backup files: 3

DUA0:[DEMO]DRAW_1.IGS;6       162  21-AUG-1990
DUA0:[DEMO]DRAW_1.IGS;5     162   7-MAY-1990
DUA0:[DEMO]DRAW_2.IGS;4       395  21-AUG-1990
DUA0:[DEMO]DRAW_2.IGS;3     395  21-AUG-1990
DUA0:[DEMO]DRAW_2.IGS;2     395   8-JUL-1990
DUA0:[DEMO]DRAW_3.IGS;2       286   8-JUL-1990

Total of 6 files, 1795 blocks
```

File names which are indented are backup files. Primary file names are not indented.

/ALLOW_ACCESS

Requests that FCX open files allowing read access by other processes..

type

Global qualifier.

format

/ALLOW_ACCESS

operations

APPEND	COMPRESS	CREATE	REFRESH	UPDATE
🟩	🟩	🟩	🟩	🟩

description

By default, FCX opens input files with exclusive access. This qualifier requests FCX to open input files allowing other processes 'read access' during compression.

example

```
$ FCX COMPRESS *.DOC/ALLOW_ACCESS (1)
```

(1) All files in the current default directory which have file type `.DOC` (highest version number only) will be selected for compression. Files will be opened allowing other processes 'read access'. The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

/ATTRIBUTE

Override default record attributes of Windows files.

type

Positional qualifier.

format

/ATTRIBUTE=option

qualifier value option

CR

Requests that the expanded files have the 'carriage return carriage control' record attribute. Each record in the expanded file will be terminated by a CR-LF combination. These characters will not explicitly reside in the file. (See /FORMAT).

NONE

Requests that the expanded files have no record attributes. If the record format is variable length (either the default for the files in question or specified using the /FORMAT qualifier), the maximum record size will be set to 512 bytes. If the fixed length record format has been specified using the /FORMAT qualifier, the maximum record size will again be set to 512 bytes.

operations

EXPAND	RETRIEVE
🟩	🟩

related qualifiers

/CONFIRM /FORMAT

description

This qualifier is used to override the default record attributes of the compressed files contained in an FCX file generated on an Windows system using DiscPac or Thoro’Pac. The default record attributes for the compressed files may be examined using the LIST/FULL operation prior to expanding the files.

note

This qualifier is ignored if the FCX file being processed was generated on a VMS system. See the section " Thoro’Pac and DiscPac Compatibility" in the Appendix.

examples

```
$ FCX EXPAND/FORMAT=VARIABLE/ATTRIBUTE=CR DOCS (1)
```

(1) The files in the transport file DOCS.FCX will be expanded in the current default directory. The files will have variable length records and will have the 'carriage return carriage control' record attribute. The expanded files will have file names that match those of the corresponding Windows files. The file types will match the Windows file extensions.

```
$ FCX EXPAND/ATTRIBUTE=NONE MEMOS (2)
```

(2) The files in the transport file MEMOS.FCX will be expanded in the current default directory. The files will have variable length records and will have no record attribute. The maximum record size will be set to 512. The expanded files will have file names that match those of the corresponding Windows files. The file types will match the Windows file extensions.

Interoperability

/BACKUP

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

type

Positional qualifier.

format

/BACKUP/BEFORE=time
/BACKUP/SINCE=time
/NOBACKUP

operations

APPEND	COMPRESS	CREATE	UPDATE
🟩	🟩	🟩	🟩

related qualifiers

[/BEFORE] [/CREATED] [/EXPIRED] [/IGNORE] [/MODIFIED] [/RECORD] [/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. This time may also be revised through use of the /RECORD option.

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

Files which have never been marked as backed up have a system time of zero recorded in the backup date field of the file's header record. The time displayed by VMS during execution of a DIRECTORY/FULL command is <No backup done>. The negated qualifier /NOBACKUP will select these files only. This negated qualifier does not require the use of a /SINCE or a /BEFORE qualifier. If one is specified, it will be ignored.

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 compressed, it is necessary to use the /IGNORE qualifier.

examples

```
$ FCX COMPRESS *.DOC/BACKUP/BEFORE (1)
```

(2) All files in the current default directory which have file type .MEM (highest version number only) and which have been marked as backed-up since 00:00 o'clock of May 12, 1986 will be selected for compression. The resultant transport file will have the same file name as the first selected file; it will have a file type of .FCX, relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS *.MEM;*/NOBACKUP (2)
```

(2) All files in the current default directory which have file type .MEM (all version numbers) and which have been marked as <No backup done> will be selected for compression. The resultant transport file will have the same file name as the first selected file; it will have a file type of .FCX, relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS *.BCK;*/IGNORE (3)
```

(3) All files in the current default directory which have file type .BCK (all version numbers) including those which have been marked as <Backups disabled> will be selected for compression. The resultant transport file will have the same file name as the first selected file; it will have a file type of .FCX, relative version number of 0 and will reside in the current default directory.

/BEFORE

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

type

Positional qualifier.

format

/BEFORE (default is TODAY)
/BEFORE=time

qualifier value time

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

BACKUP	The backup file time in the file's header
TODAY	The current day at 00:00:00.0 o'clock

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

If no time value is specified, TODAY is assumed.

operations

APPEND	COMPRESS	CREATE	EXTRACT	UPDATE
🟩	🟩	🟩	🟩	🟩

related qualifiers

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

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, /CREATED is assumed.

examples

```
$ FCX COMPRESS *.DOC/BEFORE (1)
```

(1) All files in the current default directory which have file type .DOC (highest version only) and which have been created before 00:00 o'clock of the current day will be selected for compression. The resultant transport file will reside in the current default directory; it will have the same file name as the first selected file; the file type will be .FCX and the relative version number will be 0.

```
$ FCX COMPRESS *.MEM;*/BEFORE=10:00 (2)
```

(2) All files in the current default directory which have file type .MEM (all versions) and which have been created before 10:00 o'clock of the current day will be selected for compression. The resultant transport file will have the same file name as the first selected file; it will have a file type of .FCX, relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS *.MEM;*/BEFORE=BACKUP/MODIFIED (3)
```

(3) All files in the current default directory which have file type .MEM (all versions) are candidates for compression. Of the set of candidates, files will be selected for compression which have a revision time in the file header which is earlier than the backup time for the same file. The resultant transport file will be the same as in Example 2.

/BLOCK_SIZE

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

type

Global qualifier.

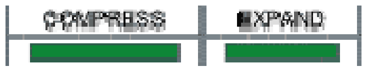
format

/BLOCK_SIZE=number

qualifier value number

The number indicates the number of bytes to be contained in a single block of a transport file. Block sizes are allowed in the range 512 to 32256 bytes.

operations



related qualifiers

[/VAR_LENGTH]

description

By default, the COMPRESS operation creates transport files as fixed length records with a record size of 512 bytes and a block size of 512 bytes. This qualifier is used to override the default record size of records in a transport file. Block sizes are allowed in the range 512 to 32256 bytes. If an invalid block size is specified, a warning message is output and compression continues with the default record size of 512.

This qualifier may be used to create a transport file with a larger block size to take advantage of a file transfer protocol which supports larger block sizes.

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

note

For magnetic tape, transport files are created as fixed length records with a default record size of 512 bytes and a default block size of 2048 bytes. Use of the /BLOCK_SIZE qualifier will affect the record size only. The block size may be overridden using the VMS MOUNT command.

By default, the EXPAND operation creates files on disk with the exact same characteristics as the original files. Files may be expanded to magnetic tape for transfer to another site. Tape blocks may vary in range and are specified using this qualifier. Tape block sizes are allowed in the range 512 to 32256 bytes. If a tape block size of 2048 is specified, the files expanded to tape will be in the same format as if the COPY utility wrote them.

examples

```
$ FCX COMPRESS/BLOCK_SIZE=2048 TEST.COM (1)
```

(1) The file `TEST.COM` in the current default directory (highest version only) will be compressed. The resultant transport file will be `TEST.FCX`. It will have a relative version number of 0 and will

reside in the current default directory. The transport file will have fixed-length records containing 2048 bytes each. The last record of the file will be padded with zeros if necessary.

```
$ FCX EXPAND/BLOCK_SIZE=2048/OUTPUT=MKA500:TEST.FCX (2)
```

(2) The transport file `TEST.FCX` in the current default directory will be expanded to the tape mounted on the device MKA500. The tape may then be transferred to another site and read using the VMS COPY utility.

/BRIEF

Requests a brief display of an FCX file.

type

Global qualifier.

format

/BRIEF (default)

operations

DIRECTORY	LIST
🟩	🟩

related qualifiers

[/ALL][/DISPLAY][/FULL][/KEY][/OUTPUT][/SUMMARY]

description

/BRIEF provides for a brief display of the contents of an FCX file to `SYS$OUTPUT`. The brief display consists of the FCX file description (if input with /COMMENT or /TITLE during compression), the version of FCX and the operating system used during compression, and the number of compressed files contained in the FCX file. The file specification and size of each compressed file is also output.

example

```
$ FCX LIST/BRIEF PARAMS (1)
```

(1) A summary listing of the contents of `PARAMS.FCX` will be displayed to `SYS$OUTPUT`. The listing will appear as follows:

```
           FCX File Compression Information

FCX file:            DISK$ZEUS:<USER>PARAMS.FCX;1
Creation date:       6-NOV-1993 23:13:35.58
FCX Version:         FCX Version 4.0-0 Level K
```

```
Operating system:  VMS Version V5.5-2
Command line:      FCX COMP/OUT=PARAMS SYS$SYSTEM:*.DAT
Created by:        USER
Created on node:   ATLAS::
Default:           DISK$ZEUS:<USER>
Algorithm:         Adapting code algorithm

SYS$SYSROOT:[SYSEXE]$.DAT;1                          81  31-AUG-1991
SYS$SYSROOT:[SYSEXE]AGEN$ADDHISTORY.DAT;1             1   4-MAR-1993
SYS$SYSROOT:[SYSEXE]AGEN$FEEDBACK.DAT;13              4   7-OCT-1993
SYS$SYSROOT:[SYSEXE]MODPARAMS.DAT;9                   4   7-SEP-1993
SYS$SYSROOT:[SYSEXE]NETCIRC.DAT;1                    21  26-AUG-1991
SYS$SYSROOT:[SYSEXE]NETCONF.DAT;1                     3  26-AUG-1991
SYS$SYSROOT:[SYSEXE]NETLINE.DAT;1                    21  26-AUG-1991
SYS$SYSROOT:[SYSEXE]NETLOGING.DAT;1                  21  26-AUG-1991
SYS$SYSROOT:[SYSEXE]NETNODE_LOCAL.DAT;1              21  26-AUG-1991
SYS$SYSROOT:[SYSEXE]NETOBJECT.DAT;1                  21  26-AUG-1991
SYS$SYSROOT:[SYSEXE]NETPROXY.DAT;1                   12  26-AUG-1991
SYS$SYSROOT:[SYSEXE]PARAMS.DAT;7                     11   4-MAR-1993
SYS$SYSROOT:[SYSEXE]SETPARAMS.DAT;12                  7   4-MAR-1993
SYS$SYSROOT:[SYSEXE]SYSALF.DAT;1                     15   5-JUL-1990
SYS$COMMON:[SYSEXE]CLUSTER_AUTHORIZE.DAT;2            1  26-AUG-1991
SYS$COMMON:[SYSEXE]JBC.DAT;1                        189  28-AUG-1991
SYS$COMMON:[SYSEXE]JBC2.DAT;1                       189  28-AUG-1991
SYS$COMMON:[SYSEXE]JBCSYSQUE.DAT;1                  108   1-JUL-1992
SYS$COMMON:[SYSEXE]LMF$LURT.DAT;2                    48   8-JUL-1992
SYS$COMMON:[SYSEXE]NETNODE_REMOTE.DAT;1              60  26-AUG-1991
SYS$COMMON:[SYSEXE]NETPROXY.DAT;1                    12  18-JUN-1990
SYS$COMMON:[SYSEXE]RIGHTSLIST.DAT;1                  30   7-SEP-1992
SYS$COMMON:[SYSEXE]SYSUAF.DAT;1                      36   7-SEP-1992
SYS$COMMON:[SYSEXE]TFF$MASTER.DAT;2                 134   8-JUL-1992
SYS$COMMON:[SYSEXE]VMS$FILE_ATTRIBUTES.DAT;1          7  17-OCT-1991
SYS$COMMON:[SYSEXE]VMSPARAMS.DAT;1                   63  20-JUL-1990

 Total of 26 files (1120 blocks)
```

/BY_RECORD

Requests that records within a file be separately expandable.

type

Global qualifier.

format

/BY_RECORD (compress and list operations)

/BY_RECORD=(option[,...]) (expand_operations)

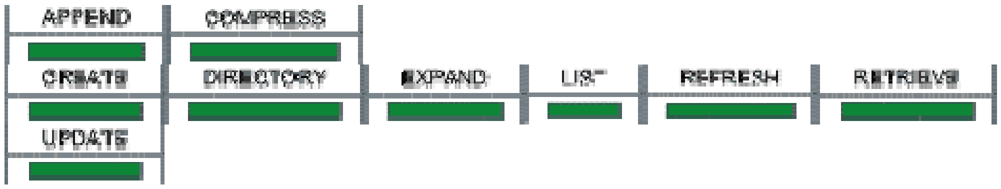
qualifier value option

The option indicates the records to be selected for expansion. If only one option is entered, the parentheses may be eliminated. The following options may be entered.

START=n	n = the beginning record to expand.
END=m	m = the last record to expand.
COUNT=n	n = number of records to expand.

If START is not specified, the first record is assumed. If neither END or COUNT is specified, the end of file is assumed.

operations



related qualifiers

description

By default, FCX maintains no separate information about records in compressed files. The file header and record separators (if present) are compressed as part of the data. This qualifier requests FCX to maintain some additional record information about the file to enable expansion of selected records. Use of this qualifier is required on compress operations (APPEND, COMPRESS, CREATE, REFRESH and UPDATE) if subsequent expansion by record is desired. The amount of information maintained by FCX is small and will have little if any impact on the compression results. The same is true of speed. Using this qualifier on compress operations will have negligible effect.

If this qualifier is specified on a list operation (LIST or DIRECTORY), the total number of records contained within each file will be displayed.

On an expand operation (EXPAND or RETRIEVE), the selected records will be expanded and either displayed (default) or directed to a file if /OUTPUT is also specified. If the selected records are being displayed and the compressed file is a binary file, the results are unpredictable.

note

/BY_RECORD is currently only supported for variable or fixed length record format files. The variable length files must not have a stream record type (STM, STMLF, or STMCR). It is ignored, if specified, for files with other file types.

note

/BY_RECORD for an expand operation (EXPAND or RETRIEVE) only operates on a single file. If a specific file is not selected, the first compressed file is selected.

examples

```
$ FCX COMPRESS/BY_RECORD TEST.COM (1)
```

(1) The file TEST.COM in the current default directory (highest version only) will be compressed, including information about records for future expansion. The resultant transport file will be TEST.FCX. It will have a relative version number of 0 and will reside in the current default directory.

```
$ FCX CREATE/BY_RECORD  COMLIB *.COM (2)
```

(2) All files in the current default directory with extension .COM (highest version only) will be compressed into a discpac COMLIB.XLB which will reside in the current default directory. Information about each file and its associated records will be included in the discpac to support future expansion of selected records.

```
$ FCX DIRECTORY/FULL/BY_RECORD COMLIB (3)
```

(3) As each file is listed in the directory of the discpac file COMLIB.XLB, total number of records contained in each file will also be included.

```
$ FCX EXPAND/BY_RECORD=(START:10,COUNT:10) TEST (4)
```

(4) The compressed file TEST.COM in the transport file TEST.FCX will be selectively expanded, i.e., ten records of the file will be expanded starting at the tenth record. The expanded records will be displayed to SYS$OUTPUT.

```
$ FCX RETRIEVE/BY_RECORD=(END:20)  COMLIB/OUT=.ZZZ (5)
```

(5) The first file contained in the discpac COMLIB.XLB, residing in the current default directory, will be selected. The first 20 records of the file will be expanded and directed to a file with extension .ZZZ and file name which matches that of the selected compressed file. The output file will reside in the current default directory and will have the same file structure as the original file.

/CBT

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

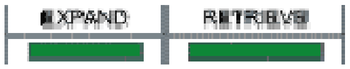
type

Positional qualifier.

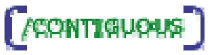
format

```
/CBT
/NOCBT
```

operations



related qualifiers


[/CONTIGUOUS]

description

/CBT (contiguous-best-try) causes the expanded 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, FCX will create an expanded file which has the same attributes. The /CBT qualifier is used to indicate contiguous attributes for other expanded files.

If there is not enough contiguous space available on the output device when FCX is creating 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

```
$ FCX EXPAND DOC/CBT (1)
```

(1) The transport file `DOC.FCX` in the current default directory is to be expanded. Each expanded 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 compressed files will still be expanded.

/COMMENT

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

type

Global qualifier

format

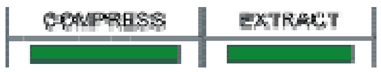
/COMMENT=string
/COMMENT="string"
/COMMENT
/NOCOMMENT (default)

qualifier value string

The supplied string is kept as a comment in the transport file. If the string is to contain 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.

operations



description

The supplied comment is kept as part of the transport file for easy identification of the contents of the file at a later time. It is used by the LIST operation in preparing displays of the contents of the transport file.

examples

```
$ FCX COMPRESS/COMMENT="this is a test" TEST.COM (1)
```

(1) The file `TEST.COM` (highest version only) in the current directory will be compressed. The contents of the comment will be included in the resultant transport file for later display by the LIST operation. The quotation marks surrounding the actual comment are necessary since the comment includes blanks.

The resultant transport file will be `TEST.FCX`. It will reside in the current default directory and have relative version number 0.

```
$ FCX COMPRESS/COMMENT=source_code *.MAR (2)
```

(2) All files in the current default directory which have file type `.MAR` (highest version only) will be compressed. The resultant transport file will contain the contents of the comment (i.e., the word "`SOURCE`"). 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 will be displayed by the LIST operation.

The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

```
$ FCX EXTRACT MYLIB *.MAR /OUTPUT=SRC/COMMENT=source (3)
```

(3) All files in the discpac file `MYLIB.XLB`, in the current default directory, which have file type `.MAR` (primary files only) will be extracted and placed into a new transport file. The created transport file, `SRC.FCX`, will contain the contents of the comment (i.e., the word "`SOURCE`"). 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 will be displayed by the LIST operation.

The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

/CONFIRM

Displays a prompt before each selected file is processed.

type

Positional qualifier.

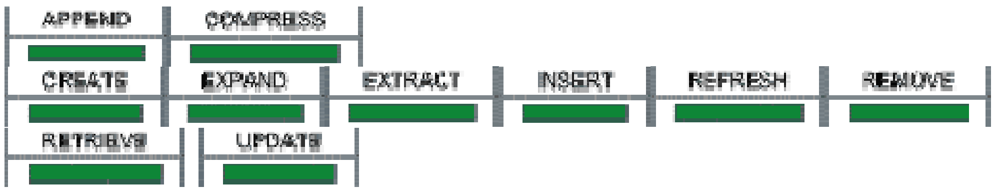
format

/CONFIRM
/NOCONFIRM (default)
/CONFIRM=OVERRIDE

qualifier value **OVERRIDE**

The **OVERRIDE** option provides additional prompts for overriding the default record format and attributes while expanding Windows files. It is only allowed if the selected operation is EXPAND or RETRIEVE and the FCX file to be expanded was generated on a Windows, Linux, or UNIX system. (See also /ATTRIBUTE and /FORMAT). The /ATTRIBUTE and /FORMAT qualifiers provide for overriding the record format and attributes of all files selected from an FCX file. **/CONFIRM=OVERRIDE** provides this on a file by file selective basis.

operations



related qualifiers

description

If the selected operation is APPEND, COMPRESS, CREATE, REFRESH, or UPDATE, a prompt is sent to `SYS$COMMAND` before each selected input file is compressed and/or deleted (if /DELETE is also specified). If the selected operation is EXPAND, EXTRACT, INSERT, or RETRIEVE, the prompt is displayed before each selected compressed file is processed.

The file name is displayed and the user must respond by typing a {Y,y} if the file is to be processed. Any other response will result in the file being skipped.

examples

```
$ FCX COMPRESS *.DOC/CONFIRM (1)
```

(1) All files in the current default directory which have file type .DOC (highest version only) are candidates for selection. The following is a sample prompt which will be displayed for each candidate file:

```
   SYS$SYSDEVICE:<USER>TEST.DOC compress? [N]
```

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

The resultant transport file will have the same file name as the first selected file; it will have a file type of .FCX, relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS *.DOC/CONFIRM/DELETE (2)
```

(2) This example is exactly the same as (1) except for the /DELETE qualifier. The user will be prompted to select files for compression as in (1) above. Once the transport file has been successfully created the user will be prompted again for the selected files only. A sample prompt is as follows:

```
   SYS$SYSDEVICE:<USER>TEST.DOC delete? [N]
```

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

The resultant file will be the same as in (1) above.

```
$ FCX EXPAND/CONFIRM DOC (3)
```

(3) The compressed files contained in the transport file DOC.FCX in the current default directory will be candidates for expansion. The user will be prompted for each file before it is expanded. The following is a sample prompt which will be displayed for each candidate file:

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

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

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

```
$ FCX EXPAND/CONFIRM=OVERRIDE TEST (4)
```

(4) The compressed files contained in the transport file TEST.FCX in the current default directory will be candidates for expansion. In this case, the transport file was generated on another system (Windows, Linux, or UNIX). The user will be prompted for each file before it is expanded. If the file is selected for expansion, the user will also be prompted for the file's record format and attributes. The following is a sample prompt sequence which will be displayed for each candidate file:

```
   C:\TEST2\TESTFILE.DOC, expand? [N]
   record format: FIXED [F]
                  STM [S],
                  STMCR [C],
                  STMLF [L],
                  VARIABLE [V]?  [V]:
```

```
    record attributes: CR [C]
                       None [N]? [C]:
```

If the user responds by typing a {Y,y} followed by a carriage return in response to the first prompt, the file will be selected for expansion and the next prompt will be displayed. If the user selects FIXED for the record format, no other prompts will be displayed. Otherwise, the third prompt will be displayed. To select the default for the file (displayed after the ?) the user need only press the RETURN key.

The selected files will be expanded in the current default directory. The file name and type will match the original. The version number will be 1.

/CONTIGUOUS

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

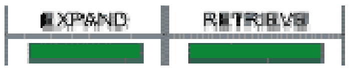
type

Positional qualifier.

format

/CONTIGUOUS
/NOCONTIGUOUS

operations



EXPAND	RETRIEVE
🟩	🟩

related qualifiers

[/CBT]

description

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

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

If there is not enough contiguous space available on the output device when FCX 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 created will not occupy consecutive blocks of storage.

example

```
$ FCX EXPAND DOC/CONTIGUOUS (1)
```

(1) The transport file DOC.FCX in the current default directory is to be expanded. Each expanded 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 and any remaining compressed files will not be expanded.

/CREATED

Requests that files be selected for compression based on the date the file was created.

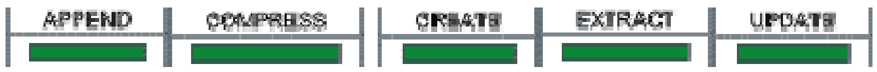
type

Positional qualifier.

format

/CREATED/BEFORE=time
/CREATED/SINCE=time

operations



related qualifiers

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

```
$ FCX COMPRESS *.DOC/BEFORE/CREATED (1)
```

(1) All files in the current default directory which have file type .DOC (highest version only) and which have been created before 00:00 o'clock of the current day will be selected for compression. The resultant transport file will have the same file name as the first selected file; it will have a file type of .FCX, relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS *.DOC/CREATED/SINCE=YESTERDAY (2)
```

(2) All files in the current default directory which have file type `.DOC` (highest version only) and which have been created since 00:00 o'clock of the previous day will be selected for compression. The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

```
$ FCX EXTRACT MYLIB *.DOC/CREATED-
/SINCE=YESTERDAY/OUTPUT=MYDOCS (3)
```

(3) All files contained in the discpac file `MYLIB.XLB` which have file type `.DOC` (primary files only) and which have been created since 00:00 o'clock of the previous day will be extracted and placed into a new transport file, `MYDOCS`.

/DELETE

Requests that input files be deleted after successful compression.

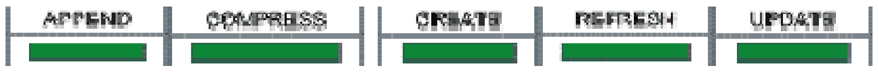
type

Positional qualifier.

format

/DELETE

operations



related qualifiers

[/CONFIRM][/VERIFY]

description

This qualifier requests that all input files be deleted after successful creation of the FCX file. Only those files which have successfully been compressed and are included in the FCX file will be candidates for deletion. 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 or /MONITOR is also specified.

Sufficient user privilege is required to delete files.

It is strongly recommended that /VERIFY be used in conjunction with /DELETE. The verify operation will be completed before any files are deleted.

If any problems are found with any file in the FCX file, no files will be deleted.

examples

```
$ FCX COMPRESS/LOG/VERIFY *.DOC/DELETE (1)
```

(1) All files in the current default directory which have file type .DOC (highest version only) will be compressed. As each file is compressed, a message is output to `SYS$OUTPUT` (see /LOG). The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

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

```
$ FCX REFRESH/DELETE/CONFIRM MYLIB (2)
```

(2) The discpac file `MYLIB.XLB` in the current default directory will be refreshed. New versions of compressed files will be added to the discpac.

A prompt will be issued prior to compression of each new file and after successful compression of the file, another prompt will be issued prior to deletion of the input files.

/DEVDIR

Requests updating of discpac files with checking of the device and directory.

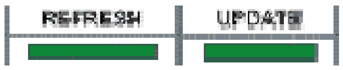
type

Global qualifier.

format

/DEVDIR (default)
/NODEVDIR

operations



description

Updating of discpac files is done by comparing those files within the discpac with those to be added. Files with the same file specification will be added as primary files and those primary files moved to backup files. If it is desired that the device and directory not be included in the name check, use /NODEVDIR. This has the effect of updating or refreshing a discpac from a directory other than the one from which it was created.

example

```
$ FCX UPDATE/NODEVDIR MYLIB *.DOC (1)
```

(1) The discpac file `MYLIB.XLB` in the current default directory is to be updated. All files with file type .DOC (highest version only) in the current default directory will be compressed and added to the discpac. If the file already exists in the discpac, it will be skipped. Files already in the discpac with the same file name and type but different date will be marked as backup files and the newly added files will become the primary files. This is done regardless of the device and directory of the files in the discpac.

/DISCPAC_DATE

Requests the date to be used for updating of discpac files be either the file creation date or file revision date.

type

Global qualifier.

format

/DISCPAC_DATE=CREATED
/DISCPAC_DATE=MODIFIED (default)

operations

CONVERT	CREATE
🟩	🟩

description

Updating of discpac files is done by comparing those files within the discpac with those to be added. The file dates of files with the same file specification will be checked. Files with the same file name and date will not be added to the discpac. This qualifier specifies which date is to be used for the date checking. The date used for creating discpac files may also be specified using the logical name FCX_DISCPAC_DATE.

example

```
$ FCX CREATE/DISCPAC_DATE=CREATED MYLIB *.DOC (1)
```

(1) The discpac file `MYLIB.XLB` will be created in the current default directory. All files with file type `.DOC` (highest version only) in the current default directory will be compressed and added to the discpac. The discpac will be marked so that future updates and refreshes will use the file creation date and time for comparisons.

/DISPLAY

Requests a modified display of an FCX file.

type

Global qualifier.

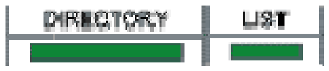
format

/**DISPLAY**[=format]

qualifier value

format=BYTES
format=BLOCKS (default)

operations



related qualifiers

[/ALL][/BRIEF][/FULL][/SUMMARY]

description

/DISPLAY provides for changing the default display of the contents of a transport or discpac. The default display provides file sizes in blocks. Using the /DISPLAY qualifier, this may be changed to provide file sizes in bytes. This is most useful for listing FCX files which were created on a different system or which contain files from another system added to a VMS transport or discpac with the APPEND or INSERT operations.

example

```
FCX LIST/DISPLAY=BYTES INC (1)
```

(1) A listing of the contents of `INC.FCX`, a transport file created on a PC, will be displayed to `SYS$OUTPUT`. The listing will be displayed with files sizes in bytes and will appear as follows:

```
          FCX File Compression Information

FCX file:              DISK$ZEUS:[USER]INC.FCX;1
Creation date:         19-DEC-1993 08:59:34.00
FCX Version:           Thoro'Pac Version 3.0-0 Level K
Operating system:      MS-DOS Version 5.0
Command line:          FCX COMP/OUT=ATLAS::[USER]INC/EXCHANGE *.INC
Default:               C:\V4SRC
Algorithm:             Adapting code algorithm

C:\V4SRC\ALGSTR.INC                        6938  11-DEC-1993
C:\V4SRC\CHECK.INC                          549  07-DEC-1993
C:\V4SRC\COLOR.INC                          614  23-OCT-1993
C:\V4SRC\COMM.INC                           730  01-DEC-1993
C:\V4SRC\COPY.INC                          1677  07-DEC-1993
C:\V4SRC\DEL.INC                            335  07-DEC-1993
C:\V4SRC\DIR.INC                            784  23-OCT-1993
C:\V4SRC\EXIT.INC                            63  27-NOV-1993
C:\V4SRC\HELP.INC                           773  07-DEC-1993
C:\V4SRC\INC.INC                            899  07-DEC-1993
```

```
C:\V4SRC\MEM.INC                                 143  05-DEC-1993
C:\V4SRC\NET.INC                                 210  11-DEC-1993
C:\V4SRC\NETNAME.INC                            2137  12-DEC-1993
C:\V4SRC\PASS.INC                                253  12-DEC-1993
C:\V4SRC\PRINT.INC                               151  12-DEC-1993
C:\V4SRC\READ.INC                                473  07-DEC-1993
C:\V4SRC\SPACE.INC                               422  16-DEC-1993
C:\V4SRC\START.INC                              1759  16-DEC-1993
C:\V4SRC\TIME.INC                                404  21-OCT-1993

 Total of 19 files (19314 bytes)
```

$ FCX DIRECTORY/DISPLAY=BLOCKS NEWINC (2)

(2) A directory of the discpac `NEWINC.XLB`, a multi-platform discpac containing both VMS and Windows files, will be displayed to `SYS$OUTPUT`. The listing will be displayed with file sizes given in blocks and appears as follows:

```
          FCX Discpac File Information

FCX file:                    DISK$ZEUS:<USER.V4SRC>NEWINC.XLB;1
Creation date:               19-DEC-1993 09:03:04.66
Date of last update:         19-DEC-1993 09:04:07.24
FCX Version:                 FCX Version 4.0-0 Level S
Operating system:            VMS Version V5.5-2
Algorithm:                   Adapting code algorithm
Update/Refresh Mode:         File Revision Date
Max Versions:                3
Number of primary files:     81
Number of backup files: 2




C:\V4SRC\ALGSTR.INC;0                        8  11-DEC-1993
C:\V4SRC\CHECK.INC;0                        12  07-DEC-1993
C:\V4SRC\COLOR.INC;0                         2  23-OCT-1993
C:\V4SRC\COMM.INC;0                          2  01-DEC-1993
C:\V4SRC\COPY.INC;0                          4  07-DEC-1993
C:\V4SRC\DEL.INC;0                           1  07-DEC-1993
C:\V4SRC\DIR.INC;0                           2  23-OCT-1993
C:\V4SRC\EXIT.INC;0                          1  27-NOV-1993
C:\V4SRC\HELP.INC;0                          2  07-DEC-1993
C:\V4SRC\INC.INC;0                           2  07-DEC-1993
DISK$ZEUS:<USER.V4SRC>MEM.INC;9              2  19-DEC-1993
C:\V4SRC\MEM.INC;-1                        1  05-DEC-1993
C:\V4SRC\NET.INC;0                         1  11-DEC-1993
C:\V4SRC\NETNAME.INC;0                       5  12-DEC-1993
C:\V4SRC\PASS.INC;0                          1  12-DEC-1993
C:\V4SRC\PRINT.INC;0                         1  12-DEC-1993
C:\V4SRC\READ.INC;0                          1  07-DEC-1993
C:\V4SRC\SPACE.INC;0                         1  16-DEC-1993
DISK$ZEUS:<USER.V4SRC>START.INC;5            3   6-DEC-1993
 C:\V4SRC\START.INC;-1                      4  16-DEC-1993
C:\V4SRC\TIME.INC;0                          1  21-OCT-1993
DISK$ZEUS:<USER.V4SRC>ALG.INC;7              1  16-DEC-1993
DISK$ZEUS:<USER.V4SRC>ANAL.INC;6             3   6-DEC-1993
DISK$ZEUS:<USER.V4SRC>APPEND.INC;5          12   6-DEC-1993
.
```

```
.
.
DISK$ZEUS:<USER.V4SRC>VERS.INC;5          7   6-DEC-1993
DISK$ZEUS:<USER.V4SRC>WRITE.INC;5        19   6-DEC-1993


Total of 83 files, 277 blocks
```

/EXCHANGE

Requests the compression defaults for FCX to insure exchange of FCX files with other systems.

type

Global qualifier.

format

/EXCHANGE

operations

COMPRESS	CREATE
🟩	🟩

related qualifiers

/GROUP /LEVEL

description

/EXCHANGE is a shorthand way to set defaults to insure that all compressed files may be expanded by DiscPac orThoro'Pac on a Windows system. Once an FCX file is created using /EXCHANGE, the defaults will remain in effect for that file with future operations (e.g., APPEND, REFRESH, or UPDATE). This applies to both transport files and discpac files.

The defaults used are as follows:

/LEVEL=0

/EXCHANGE sets the defaults to insure that all compressed files may be expanded by Thoro'Pac or DiscPac.

example

```
$ FCX COMPRESS/EXCHANGE *.DOC (1)
```

(1) All files in the current default directory with file type .DOC (highest version only) will be compressed using the adapt code algorithm with the exchange parameters set. The resultant

transport file will have the same name as the first file compressed; the file type will be .FCX and it will reside in the current default directory.

/EXCLUDE

Requests that the specified files be excluded during file selection.

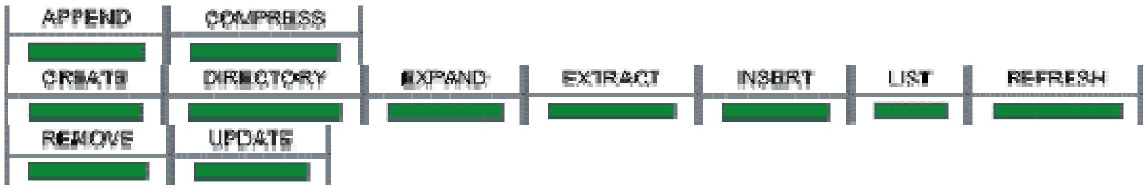
type

Positional qualifier.

format

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

operations



related qualifiers

[/SELECT]

description

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; the device and directory default to the current process defaults. The version number will default to * (i.e., all versions). 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.

examples

```
$ FCX COMPRESS *.*;*/EXCLUDE=(*.COM,*.DAT) (1)
```

(1) All files in the current default directory will be selected for compression with the exception of any file which has file type `.COM` or file type `.DAT`. The resultant transport file will have the same file name as that of the first input file included in the compression, i.e., the first file found in the current directory that does not have file type `.COM` or `.DAT`. The transport file will have a file type `.FCX,` relative version number of 0, and will reside in the current default directory.

```
$ FCX APPEND *.COM;*/EXCLUDE=LOGIN.COM TEST (2)
```

(2) All files in the current default directory which have file type `.COM` (all versions) will be compressed with the exception of any file which has file name `LOGIN`. These files will be appended to the transport file, `TEST.FCX` which resides in the current default directory.

```
$ FCX EXPAND/EXCLUDE=(*.COM,*.DAT) TEST.FCX (3)
```

(3) All compressed files contained in the transport file `TEST.FCX` in the current default directory will be expanded with the exception of any file which has file type `.COM` or file type `.DAT`. The expanded files will reside in the current default directory. The file names, types and version numbers will match those of the compressed files.

```
$ FCX EXPAND TEST.FCX/EXCLUDE=LOGIN.COM-
$_ /SELECT=*.COM (4)
```

(4) All compressed files contained in the transport file `TEST.FCX` in the current default directory which have file type `.COM` (all versions) will be expanded with the exception of any file which has filename `LOGIN`. The expanded files will reside in the current default directory. The file names, types and version numbers will match those of the compressed files.

/EXPIRED

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

type

Positional qualifier.

format

/EXPIRED/BEFORE=time
/EXPIRED/SINCE=time
/NOEXPIRED

operations

APPEND	COMPRESS	CREATE	UPDATE
🟩	🟩	🟩	🟩

related qualifiers

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

description

/EXPIRED causes files to be selected from the input-file-spec based upon the system time recorded in the expired 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 using a `SET FILE` 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 the /BEFORE or /SINCE will result in these files never being selected for compression. 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

```
$ FCX COMPRESS *.MEM/BEFORE/EXPIRED/DELETE (1)
```

(1) All files in the current default directory which have file type `.MEM` (highest version only) and which have an expiration time before 00:00 o'clock of the current day will be selected for compression. After successful creation of the transport file, all input files will be deleted.

The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS *.DOC;*/SINCE=TODAY/EXPIRED (2)
```

(2) All files in the current default directory which have file type `.DOC` and which have an expiration time later than 00:00 o'clock of the current day will be selected for compression. The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS *.DOC;*/NOEXPIRED (3)
```

(3) All files in the current default directory which have file type `.DOC` and which have no expiration time will be selected for compression. The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

/FAIL

Requests FCX to terminate processing if it encounters any errors.

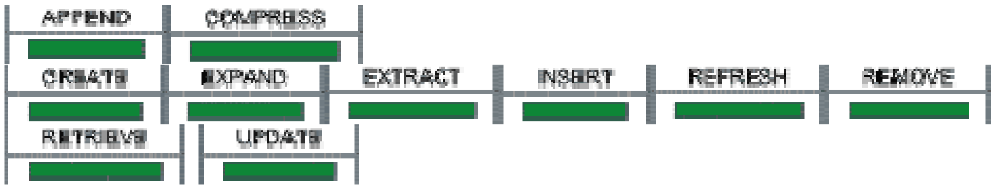
type

Global qualifier.

format

/FAIL

operations



description

By default, FCX continues processing files even if an error has occurred and been reported. This qualifier requests FCX to terminate processing after an error has occurred. This is most often used during restoration of files (EXPAND, RETRIEVE) but may be used on any operation.:

example

```
$ FCX EXPAND/FILE TEXTFILES  (1)
```

(1) All files in the transport file TEXTFILES.FCX in current default directory will be expanded. FCX will terminate processing on a non-fatal error. This may occur if a file to be expanded already exists. This could also occur if the disk is full. The appropriate error message will have been reported.

/FORMAT

Override default record format of Windows files.

type

Positional qualifier.

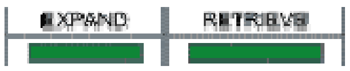
format

/FORMAT=option

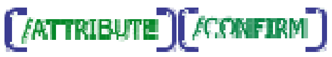
qualifier value option

See description.

operations



related qualifiers



description

This qualifier is used to override the default record format of the compressed files contained in an FCX file generated on a Windows, UNIX, or Linux system. The default record format of the compressed files may be examined using the LIST/FULL or DIRECTORY/FULL operation prior to expanding the files. The following options are available:

BINARY

Requests that the expanded files have stream (STM) record format. The record attributes will be set to 'carriage return carriage control'. No interpretation of the file data is done.

FIXED

Requests that the expanded files have fixed length record format. This format is incompatible with the record attribute of 'carriage return carriage control'. (See /ATTRIBUTE). All records in the expanded files will have a record size of 512 bytes.

STM

Requests that the expanded files have stream (STM) record format. The record attributes will be set to 'carriage return carriage control' and each record will be delimited by a carriage return-line feed (CR/LF) combination.

STMCR

Requests that the expanded files have stream carriage return (STMCR) record format. The record attributes will be set to 'carriage return carriage control' and each record will be delimited by a carriage return (CR).

STMLF

Requests that the expanded files have stream line feed (STMLF) record format. The record attributes will be set to 'carriage return carriage control' and each record will be delimited by a line feed (LF).

VARIABLE

Requests that the expanded files have variable length records. If the record attributes are 'carriage return carriage control', each record will be delimited by a carriage return-line feed combination. These characters are not copied to the expanded file. The size of the records will vary. If the record attribute has been set to None (See /ATTRIBUTE) the maximum record size will be set to 512. In this case, each record will be exactly 512 bytes long, except for the last record. The last record will be long enough to complete the file.

note

This qualifier is ignored if the FCX file being processed was generated on a VMS system. See the section " Thoro'Pac and DiscPac Compatibility" in the Appendix.

examples

```
$ FCX EXPAND/FORMAT=VARIABLE/ATTRIBUTE=CR TEST (1)
```

(1) The files in the transport file TEST, which was generated on a Windows system using DiscPac, will be expanded in the current default directory. The files will have variable length

records and will have the 'carriage return carriage control' record attribute. The file names and types of the expanded files will match the file names and extensions of the corresponding Windows files. The version numbers will be 1.

```
$ FCX EXPAND/FORMAT=FIXED MEMO (2)
```

(2) The files in the transport file MEMO, which was generated on a Windows system using Thoro'Pac, will be expanded in the current default directory. The files will have fixed length records and will have no record attribute. The maximum record size will be set to 512. If the last record in the file is less than 512 bytes, it will be padded with zeros to make a complete 512 byte record. The file names and types of the expanded files will match the file names and extensions of the corresponding Windows files. The version numbers will be 1.

/FULL

Requests a detailed display of an FCX file.

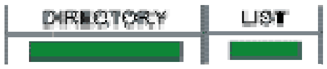
type

Global qualifier.

format

```
/FULL
```

operations



related qualifiers

/ALL
/BRIEF
/DISPLAY
/KEY
/OUTPUT
/SUMMARY

description

/FULL provides for a complete display of the contents of one or more FCX files to SYS$OUTPUT. The display consists of the FCX file description (if input with /COMMENT or /TITLE during compression), the version of FCX and the operating system used during compression, and the number of compressed files contained in the FCX file. The file specification of each compressed file contained in the FCX file as well as additional information from the file header is also provided.

example

```
$ FCX LIST/FULL SYSUAF (1)
```

(1) Sample output, as follows, will be displayed to SYS$OUTPUT:

```
          FCX File Compression Information

FCX file:             DISK$THOR:<ANN.V4SRC>VAXREF.FCX;1
Creation date:         7-NOV-1993 00:37:53.53
FCX Version:          FCX Version 4.0-0 Level K
Operating system:     Alpha VMS Version V1.5
Command line:         FCX COMP VAXREF.GEN
Created by:           ANN
Created on node:      THOR::
Default:              DISK$THOR:<ANN.V4SRC>
Algorithm:            Adapting code algorithm

DISK$THOR:<ANN.V4SRC>VAXREF.GEN;2                            Size:
   151/153
System:    Alpha VMS
Algorithm: Adapting code algorithm
Created:  1-NOV-1993 14:43
Revised:  1-NOV-1993 14:43 (5)
Backup:   <No backup done>
Expires:  <None specified>
File organization:  Sequential
File attributes: Allocation = 153, Extend = 0
Global buffer count = 0, No version limit, Contiguous Best Try
Record format:  Stream
Record attributes:     Carriage return
File protection: System:RWED, Owner:RWED, Group:, World:
Owner:  [USER]

Total of 1 file (151 blocks)
```

GROUP

Specifies that the files to be compressed be treated as one file. This usually leads to much better compression.

type

Global qualifier.

format

/GROUP
/NOGROUP (default)

operations

COMPRESS

related qualifiers

/LEVEL

description

By default, each file to be compressed is taken as an entity. This qualifier requests that all the files to be compressed be taken as a group, or bound together. The result is usually much better compression, from 10% to 20%, depending on the files.

The only drawback to using this qualifier is the loss of selective expansion of files. If used, each file must be expanded before the next; all files must be expanded in memory. If it is desired to expand all files, such as after a file transfer, then the positive results from the added compression are considerable. If the transport to be kept online for future selective expansion of files, then it is probably best not to use this qualifier, especially if the files are large.

This qualifier is most effective when used on a large number of small to medium files.

note

This qualifier requires /LEVEL=1 and automatically sets it.

example

```
$ FCX COMPRESS/GROUP *.COM (1)
```

(1) All of the files with extension .COM in the current default directory (highest version only) will be compressed using Level 1 compression with /GROUP The resultant transport file will be TEST.FCX. It will have a relative version number of 0 and will reside in the current default directory.

/IGNORE

Requests that FCX ignore certain file processing restrictions or certain error messages.

type

Positional qualifier.

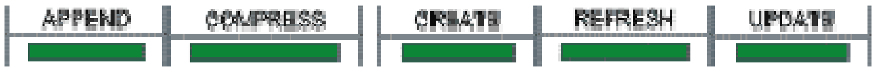
format

/IGNORE=option
/IGNORE (default=NOBACKUP)

qualifier value option

See description.

operations



related qualifiers

/BACKUP

description

Processing restrictions or messages may be ignored by FCX by using the following options:

HIGHER

By default, FCX generates an error message if a file with a higher version than the one it is expanding already exists. This can happen if you are expanding a transport file containing files with multiple version. The one with the highest version gets expanded first. This qualifier option instructs FCX to ignore this condition and expand all the files without generating any warning messages.

INTERLOCK

By default, FCX will not compress files which are currently open by another process or user. This option can be used for 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, FCX 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.)

SHORT

Files which have fixed length records sometimes have a last record which is shorter than the others. FCX compresses only the data portion of the last record and issues a warning message. Once expanded, the new file and the original file will be identical up to the EOF, but the VMS DIFFERENCES utility may indicate that the files do not compare. This is due to the part of the last record which extends beyond the end of file mark; DIFFERENCES does a block by block comparison of fixed length record files. This anomaly may also be observed using the VMS COPY utility and then DIFFERENCES.

FCX will generate warning messages when the above condition occurs. If /IGNORE=SHORT is specified, FCX will suppress the warning messages. These messages are only generated for evaluation kits. If the version of FCX being used is a production kit, these warning messages are suppressed by default. If production kit, these warning messages are suppressed by default. If it is desired to see these messages use /**IGNORE=NOSHORT**.

note

If INTERLOCK or [NO]SHORT is specified, NOBACKUP must be explicitly specified if it is desired.

example

```
$ FCX COMPRESS *.DOC/IGNORE (1)
```

(1) All files in the current default directory which have file type .DOC (highest version number only) will be selected for compression. Files which have the file attribute 'Backups disabled' will be compressed. The resultant transport file will have the same file name as the first selected file; it will have a file type of .FCX, relative version number of 0 and will reside in the current default directory.

/KEY

Specifies an identification string associated with a group of compressed files contained within a discpac file.

type

Global qualifier.

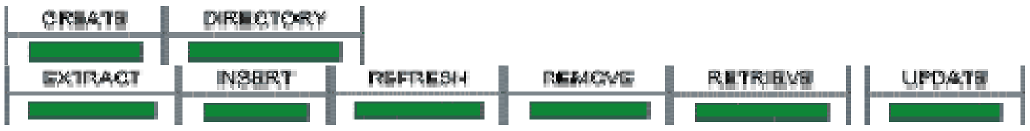
format

/KEY=string
/KEY="string"
/KEY
/NOKEY (default)

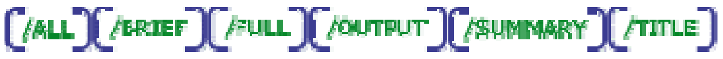
qualifier value string

The supplied string is kept as an identifier associated with a group of files in the discpac file. If the string is to contain blank and/or non-alphanumeric characters, it must be enclosed in quotation marks. The maximum length of a key is 80 characters including the quotation marks if any.

operations



related qualifiers



description

Key strings are used to identify groups of files within an FCX discpac file. The key specified will be associated with the group of files specified for the particular operation requested.

/KEY is used during the CREATE operation to identify the files specified in the input-file-spec. The key will remain with this group of files as long as they are in the discpac. The same is true with the UPDATE and REFRESH operations. In each of these cases, the string is required.

/KEY is used during the DIRECTORY, REMOVE, and RETRIEVE operations to select files from the discpac. For the REMOVE and RETRIEVE operations, the string is required. The string is optional for the DIRECTORY operation. If not specified, the displayed directory will contain the key string for each file.

examples

```
$ FCX CREATE/KEY=VER_1 DOCLIB *.DOC (1)
```

(1) All files in the current default directory with file type .DOC will be compressed (highest version only) and included in the discpac file DOCLIB.XLB which will also reside in the current default directory. Each of the .DOC files included in the discpac will be associated with the key string VER_1.

```
$ FCX UPDATE/KEY="Version 2" DOCLIB  [MYDIR]*.DOC (2)
```

(2) All files in the directory MYDIR having file type .DOC (highest version only) will be compressed and added to the discpac DOCLIB.XLB in the current default directory. Each of the newly added files will be associated with the key string "Version 2".

```
$ FCX RETRIEVE/KEY=VER_1 DOCLIB *.DOC (3)
```

(3) The compressed files having file type .DOC in the FCX discpac file DOCLIB which resides in the current default directory and which are associated with the key string VER_1 are selected for expansion. The expanded files will reside in the current default directory and will have file names which match those of the compressed files.

```
$ FCX DIRECTORY/KEY DOCLIB (4)
```

(4) A directory of the discpac file DOCLIB.XLB which resides in the current default directory will be displayed. The key of each file will be included in the display.

```
$ FCX DIRECTORY/FULL/KEY="Version 2" DOCLIB (5)
```

(5) A DIRECTORY display of the contents of the discpac file DOCLIB.XLB in the current default directory will be generated. (See /FULL for a sample of the contents of the display). Only those files whose key string is "Version 2" will be included in the display.

```
$ FCX REFRESH/KEY=VER_3 DOCLIB (6)
```

(6) All files in the discpac file DOCLIB.XLB in the current default directory will be refreshed. The key string associated with any newly added files will be VER_3. If the maximum number of versions is exceeded for any file, the oldest version of that file will be removed from the discpac.

/LEVEL

Requests the compression level.

type

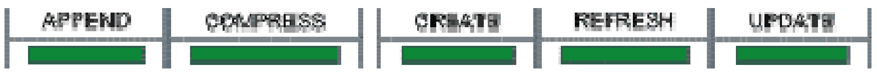
Global qualifier.

format

/LEVEL=0 (default)

/LEVEL=1 (maximum compression)

operations



APPEND	COMPRESS	CREATE	REFRESH	UPDATE
🟩	🟩	🟩	🟩	🟩

description

The /LEVEL qualifier selects the compression algorithm. Level 0 gives good compression for most files and is the fastest. With Level 1 you may save several thousand more blocks, depending upon the size of the input files. It will however take longer to compress. Level 0 is the default if no level is selected.

example

```
$ FCX COMPRESS/LEVEL=1 *.DOC (1)
```

(1) All files in the current default directory with file type .DOC (highest version only) will be compressed using the maximum algorithm. This will achieve more compression but be a bit slower. The resultant transport file will have the same name as the first file compressed; the file type will be FCX and it will reside in the current default directory.

/LICENSE

Loads the specified FCX product license file as a logical name.

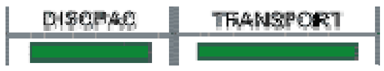
type

Global qualifier.

format

/LICENSE

operations



DISCPAC	TRANSPORT
🟩	🟩

description

Requests that the FCX product license file be loaded as the logical name FCX_DISCPAC_LICENSE or FCX_TRANSPORT_LICENSE.   The logical name will remain in tact until the system is restarted, thus this operation need be done once each time the system is restarted. The logical name equivalence string which is generated is in binary format and may not be defined in any other manner.

example

```
$ FCX TRANSPORT/LICENSE (1)
```

(1) The FCXtransport license file FCX_TRANSPORT.LICENSE, located in SYS$MANAGER, is read and a binary equivalence string created for the logical name FCX_TRANSPORT_LICENSE

Loading the License Files

.

/LOG

Display progress and status messages to SYS$OUTPUT or a log file.

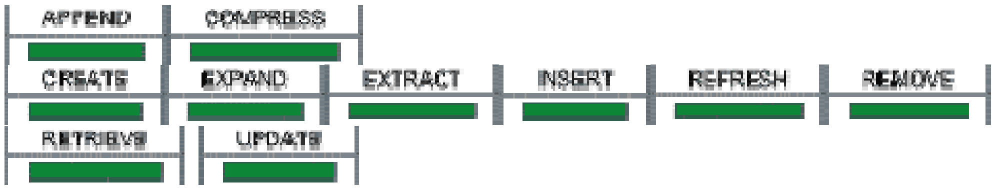
type

Global qualifier.

format

LOG
/LOG=log-file-spec
/NOLOG (default)

operations



APPEND	COMPRESS				
🟩	🟩				
CREATE	EXPAND	EXTRACT	INSERT	REFRESH	REMOVE
🟩	🟩	🟩	🟩	🟩	🟩
RETRIEVE	UPDATE				
🟩	🟩				

related qualifiers

/MONITOR /STATISTICS /VERIFY

description

Displays the file specification of each input file processed as the command executes; by default, the display is written to SYS$OUTPUT. If a log-file-spec is provided, the display is written to the file. The log-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 .LOG; the default version will be 0. There is no default for the file name; the user must supply one.

If the /MONITOR qualifier is also present and a log-file-spec has been specified, the display is sent to the file and the current SYS$OUTPUT device as well.

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

examples

```
$ FCX COMPRESS/LOG SYS$SYSTEM:SYSUAF.DAT (1)
```

(1) The file SYSUAF.DAT in the directory pointed to by the logical name SYS$SYSTEM will be compressed. The resultant transport file will reside in the current default directory. The file will be SYSUAF.FCX with version 0. The following is a sample of the information which is sent to SYS$OUTPUT during execution of this command:

```
%FCX-S-COMPRESSED,  SYS$SYSROOT:[SYSEXE]SYSUAF.DAT;2 compressed (34
blocks)
```

The number of blocks output indicates the number of input blocks to the COMPRESS operation, i.e., the number of blocks in the original file.

```
$ FCX COMPRESS/LOG=LOGFILE *.DAT;* (2)
```

(2) All files in the current default directory which have file type .DAT will be compressed. A log file called LOGFILE.LOG will be created in the current default directory. It will have relative version 0. The contents of the log file will be similar to the output shown in (1) above but will have additional information about the process creating the transport file. A sample is shown below.

```
FCX Version 4.0 Level K
Copyright 1988-1993 Innovative Computer Systems, Inc.
All rights reserved.

 %FCX-I-CMDLINE,  FCX COMPRESS/LOG=LOGFILE *.DAT;*
-FCX-I-FCXTIME, Execution time is 4-JUN-1993 22:24:22.21
-FCX-I-DEFAULT, Default directory is SYS$SYSROOT:[SYSEXE]

%FCX-S-COMPRESSED,  SYS$SYSROOT:[SYSEXE]MODPARAMS.DAT;2 compressed (2
blocks)
%FCX-S-COMPRESSED,  SYS$SYSROOT:[SYSEXE]NETCIRC.DAT;1 compressed (20
blocks)
%FCX-S-COMPRESSED,  SYS$SYSROOT:[SYSEXE]NETCONF.DAT;1 compressed (2
blocks)
.
.
.

%FCX-S-COMPRESSED,  SYS$SYSROOT:[SYSEXE]SYSUAF.DAT;2 compressed (34
blocks)
%FCX-S-COMPRESSED,  SYS$SYSROOT:[SYSEXE]VMSMAIL.DAT;1 compressed (10
blocks)
```

The resultant transport file will have the same file name as the first selected file; it will have a file type of .FCX, relative version number of 0 and will reside in the current default directory.

/MAX_VERSIONS

Specifies the maximum number of versions of a file allowed in a discpac file.

type

Global qualifier.

format

/MAX_VERSIONS=number

qualifier value

number

The number indicates the maximum number of versions of a file which will be allowed in a discpac file. The default number is 10. The maximum number allowed is 100.

operations

CREATE

description

FCX discpac files contain primary and backup files in compressed format. The primary file refers to the latest version of a file. All other versions of the same file are called backup files. The total number of versions of a file is governed by the value specified with /MAX_VERSIONS. If the value is zero, there is no version limit on the files within a discpac.

Once a discpac has been updated or refreshed several times, the number of backup files (plus one for the primary file) may grow to reach the maximum number as specified by this qualifier. When this happens, the oldest version of the file will be removed from the discpac.

example

```
$ FCX CREATE/MAX_VERSIONS=5 MYLIB *.COM (1)
```

(1) All files having file type .COM in the current default directory (highest version only) will be compressed and included in the discpac file MYLIB.XLB which will also reside in the current default directory. Files in the discpac may be refreshed or updated until the total number of versions of any given file reaches 5. Once that number is exceeded for any given file, the oldest version of that file will be removed from the discpac.

/MODIFIED

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

type

Positional qualifier.

format

/MODIFIED/BEFORE=time
/MODIFIED/SINCE=time

operations

APPEND | COMPRESS | CREATE | EXTRACT | UPDATE

related qualifiers

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

```
$ FCX COMPRESS [...]*.*/BEFORE/MODIFIED (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 selected for compression. The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS *.*/SINCE=(5-JAN-1987:9:00)/MODIFIED (2)
```

(2) All files in the current default directory (highest version only) which have a revision time later than 09:00 o'clock of January 5,1987 will be selected for compression. The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

/MONITOR

Request informational messages be displayed as the operation executes.

type

Global qualifier.

format

/MONITOR
/NOMONITOR (default)

operations

APPEND	COMPRESS				
🟩	🟩				
CREATE	EXPAND	REFRESH	REMOVE	RETRIEVE	UPDATE
🟩	🟩	🟩	🟩	🟩	🟩

related qualifiers

[/LOG] [/STATISTICS] [/VERIFY]

description

This qualifier provides for additional output (see /LOG) to `SYS$OUTPUT`. If /LOG is also specified and a log file given, messages will be displayed to `SYS$OUTPUT` as well as the log file.

During the EXPAND and RETRIEVE operations, the name of each selected compressed file is displayed as it is selected and expanded.

During the DIRECTORY and LIST operations, if /VERIFY is also selected, the name of each selected compressed file is displayed as it is verified.

This qualifier has the same effect as /LOG for all other operations. It supersedes the use of /LOG unless a log-file-spec is specified.

example

```
$ FCX COMPRESS/MONITOR/LOG=LOGFILE [...]*.*;* (1)
```

(1) All files in the directory tree starting at the current default directory will be compressed. During the operation, messages will be output to both `SYS$OUTPUT` and the log file `LOGFILE.LOG` (see /LOG) indicating when each file is compressed. The log file (`LOGFILE.LOG`) will reside in the current default directory.

The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, relative version number of 0 and will reside in the current default directory.

/NEW_VERSION

Requests that expanded files with matching file names have a new (higher) version number.

type

Positional qualifier.

format

/NEW_VERSION
/NONEW_VERSION (default)

operations

EXPAND	RETRIEVE
🟩	🟩

related qualifiers

/REPLACE

description

/NEW_VERSION allows identically named files to be expanded, setting the version number of the expanded file to the highest existing number plus one. By default, FCX will not expand 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 is another.

By default, expansion of files results in the expanded files having the same version number as the compressed file. If an identically named file exists with a higher version number, the compressed file is expanded (with the lower version number) and a warning message is generated. The use of /NEW_VERSION will give the newly expanded 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.

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

example

```
$ FCX EXPAND DOC,MEM/NEW_VERSION (1)
```

(1) The compressed files contained in the transport files DOC.FCX and MEM.FCX in the current default directory will be expanded. If existing files with the same version number exist for the files contained in DOC.FCX, the compressed files will not be expanded. Error messages will be generated. If existing files with the same version number exist for the files contained in MEM.FCX, the compressed files will be expanded and given a higher version number. All expanded files will reside in the current default directory and have the same file name and type as the corresponding compressed files.

/OUTPUT

Specifies the output device, directory and file name for the output file(s) generated by the operation.

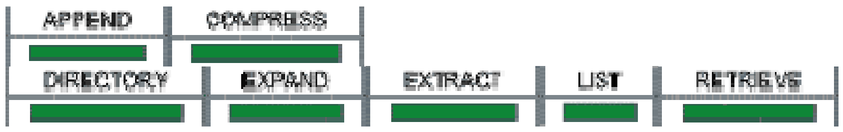
type

Global qualifier.

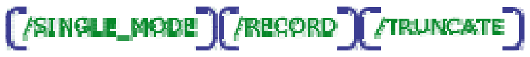
format

/OUTPUT=output-file-spec

operations



related qualifiers



description

The output-file-spec is required in all uses of this qualifier. The definition of what the output-file-spec contains is different for each operation.

COMPRESS

/OUTPUT is used during the COMPRESS operation to define the destination of the resultant transport file. The default file name of the transport file is the same as the first input file selected for compression. It will have an extension of `.FCX` and will reside in the current process default directory on the current default device.

This qualifier is used to override the default output-file-spec for the transport file. This is especially useful if the input-file-spec contains wild-cards in which case the default FCX filename would match that of the first input file and may have very little meaning.

The output-file-spec may be any valid VMS file specification, including the network node of a DECnet node to which the transport file will be sent during compression.

Use of this qualifier is required if the transport file is to reside on a different device from the current device, e.g., if the transport file is to be written to tape or to a different node.

EXPAND or RETRIEVE

/OUTPUT is used during the EXPAND or RETRIEVE operations to override the default expanded file name(s). The default device and directory are the current process defaults. The default expanded file name matches that of the original, including file type and version. If the output-file-spec contains a file name or a file type, the expanded file version number will default to relative version 0.

This qualifier is useful for renaming files as they are expanded. **It is required if a compressed directory tree is to be expanded as a directory tree.**

LIST or DIRECTORY

/OUTPUT is used during the LIST or DIRECTORY operation to cause the LIST display (see /BRIEF and /FULL) to be written to a file; by default, the display is written to `SYS$OUTPUT`.

The default file type of the output file is .LIS; the default file name is the same as the first file specified in the input-file-spec; the default version number is 0; the default device and directory are the current process defaults.

EXTRACT

/OUTPUT is required for the EXTRACT operation to specify the file name of the transport file to contain the extracted files.

examples

```
$ FCX COMPRESS/OUTPUT=DOCS *.DOC;* (1)
```

(1) All files in the current default directory with file type .DOC will be compressed (all version numbers). The resultant transport file will be called DOCS.FCX. It will reside in the current default directory and will have relative version number 0.

```
$ FCX COMPRESS/OUTPUT=RNODE::MYDIR *.* (2)
```

(2) All files in the directory will be compressed (highest version only). The resultant transport file will be called MYDIR.FCX. It will reside on the DECnet node RNODE in the current default directory and will have relative version number 0.

```
$ FCX EXPAND OLD_C_FILES/SELECT=*.C/OUT=*.OLD_C (3)
```

(3) The compressed files having file type .C in the FCX file OLD_C_FILES.FCX which resides in the current default directory are selected for expansion. As each file is expanded, it will be given the file type .OLD_C. The expanded files will reside in the current default directory and will have file names which match those of the compressed files. The expanded files will have version 0. This is useful for comparing the old and the new files.

```
$ FCX EXPAND MYDIR/OUTPUT=[...] (4)
```

(4) The transport file MYDIR.FCX residing in the current default directory was created from a directory tree. The user now wishes to create the same directory tree starting at the current default directory. All compressed files will reside in the new directory tree following the same structure as the original directory tree. New directories will be created by the operation as required. The file names, types and version numbers of the expanded files will be the same as the compressed files.

If the user had not requested an output-file-spec different from the default, all the compressed files would have been expanded in the current default directory. The original directory structure would not have been maintained.

```
$ FCX LIST/FULL/OUTPUT=MOD MODPARAMS (5)
```

(5) A LIST display of the contents of the file MODPARAMS.FCX in the current default directory will be generated. (See /FULL for a sample of the contents of the display). The display will be written to the file MOD.LIS in the current default directory.

```
$ FCX EXTRACT MYLIB *.DOC /OUTPUT=DOCFCX (6)
```

(6) The primary version of each file with file type .DOC residing in the discpac MYLIB.XLB in the current default directory will be extracted. The extracted files will be written to a new transport file

call DOCFCX.FCX, also in the current default directory. No compression or expansion of files is done.

/OWNER_UIC

Request expanded files with the indicated UIC.

type

Positional qualifier.

format

/OWNER_UIC=[uic]

qualifier value [uic]

The UIC may be specified as an identifier or in standard UIC format as described in the "OpenVMS DCL Concepts Manual". The []s are required. Numbers must be input in octal.

operations

EXPAND	RETRIEVE
🟩	🟩

related qualifiers

[/ACL]

description

By default, FCX expands 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 /OWNER_UIC 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

```
$ FCX EXPAND DOC/OWNER_UIC=[USER]  (1)
```

(1) The compressed files contained in the transport files DOC.FCX in the current default directory will be expanded with the UIC of [USER]. All expanded files will reside in the current default directory and have the same file name and type as the corresponding compressed files.

/PASSWORD

Specifies a password for the resultant FCX file.

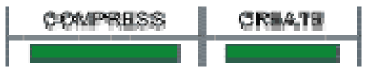
type

Global qualifier.

format

/PASSWORD

operations



description

This qualifier is used to add password protection to an FCX file. The user will be prompted to enter the password once execution has begun. 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 APPEND, DIRECTORY, LIST, EXPAND, EXTRACT, INSERT, REFRESH, REMOVE, RETRIEVE and UPDATE operations.

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

example

```
$ FCX COMPRESS/PASSWORD TEST.COM (1)
```

(1) The file `TEST.COM` in the current directory will be compressed. 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 resultant transport file `TEST.FCX`. Using other operations requires knowledge of this password. It MUST be entered for any other operations to proceed.

/PRIORITY

Requests the FCX process to execute at the specified priority.

type

Global qualifier.

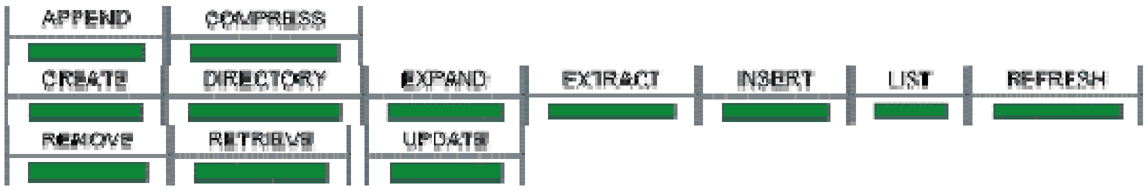
format

/PRIORITY=number

qualifier value number

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

operations



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.

example

```
$ FCX COMPRESS/PRIORITY=3/OUTPUT=ALLDISK  [*...]*.*;* (1)
```

(1) All files on the current default disk will be candidates for compression. The process will set its priority level to 3. The resultant transport file will be called `ALLDISK.FCX` and will reside in the current default directory.

/REBUILD

Rebuilds the internal index of the specified discpac file. This is necessary to upgrade a discpac from one internal level to the next.

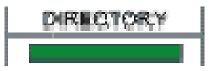
type

Global qualifier.

format

/REBUILD

operations



related qualifiers

[/VERIFY]

description

This qualifier provides for a rebuild of the internal index of a discpac file. It is used to update a discpac from one level to the next (e.g., libraries built with Version 5.x of FCX were Level T libraries; libraries built with Version 6.0 are Level U). Most operations perform the updating automatically, the exception being the EXTRACT, INSERT, and REMOVE operations. This update may also be performed using a DIR/REBUILD operation.

example

```
$ FCX DIRECTORY/REBUILD MYLIB (1)
```

(1) The discpac file `MYLIB` in the current default directory will be rebuilt and upgraded to the most current internal level.

/RECORD

Requests that processed files have the backup date field in the file header updated to the current time. This will be the time of creation of the transport file and all files marked will have the same time.

type

Positional qualifier.

format

/RECORD
/RECORD=0
/NORECORD (default)

qualifier value 0

Requests that the backup date field of the expanded files be set to zero, indicating the file is not backed up. This value is only allowed with the EXPAND operation.

operations

APPEND	COMPRESS	EXPAND

related qualifiers

/BACKUP /OUTPUT

description

By default, FCX will not change anything in a file header. The sole exception is the backup date field. /RECORD causes FCX to record the current date-time in the backup time field of each selected file header record. This provides for compressed files to be marked as backed up.

This field is also used by the VMS BACKUP Utility. (See the "OpenVMS BACKUP Utility Reference Manual" for details).

Sufficient user privilege is required.

examples

```
$ FCX COMPRESS/RECORD/OUTPUT=DKA500: *.*;* (1)
```

(1) All files in the current default directory will be compressed. Each file will be updated to have its backup date field reflect the time of compression.

The resulting transport file will reside on device DKA500. The transport file name will match that of the first file compressed and the file type will be `.FCX`.

```
$ FCX EXPAND/RECORD=0 DOC (2)
```

(2) The compressed files contained in the transport file `DOC.FCX` will be expanded. Each expanded file will be updated to have its backup date field set to zero to indicate the file has never been backed up.

All expanded files will reside in the current default directory and have the same file name and type as the corresponding compressed files.

/RECORD

Requests that processed files have the backup date field in the file header updated to the current time. This will be the time of creation of the transport file and all files marked will have the same time.

type

Positional qualifier.

format

/RECORD
/RECORD=0
/NORECORD (default)

qualifier value 0

Requests that the backup date field of the expanded files be set to zero, indicating the file is not backed up. This value is only allowed with the EXPAND operation.

operations

APPEND	COMPRESS	EXPAND
🟩	🟩	🟩

related qualifiers

/BACKUP /OUTPUT

description

By default, FCX will not change anything in a file header. The sole exception is the backup date field. /RECORD causes FCX to record the current date-time in the backup time field of each selected file header record. This provides for compressed files to be marked as backed up.

This field is also used by the VMS BACKUP Utility. (See the "OpenVMS BACKUP Utility Reference Manual" for details).

Sufficient user privilege is required.

examples

```
$ FCX COMPRESS/RECORD/OUTPUT=DKA500: *.*;* (1)
```

(1) All files in the current default directory will be compressed. Each file will be updated to have its backup date field reflect the time of compression.

The resulting transport file will reside on device DKA500. The transport file name will match that of the first file compressed and the file type will be `.FCX`.

```
$ FCX EXPAND/RECORD=0 DOC (2)
```

(2) The compressed files contained in the transport file `DOC.FCX` will be expanded. Each expanded file will be updated to have its backup date field set to zero to indicate the file has never been backed up.

All expanded files will reside in the current default directory and have the same file name and type as the corresponding compressed files.

/REPLACE

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

type

Positional qualifier.

format

/REPLACE
/NOREPLACE (default)

operations

EXPAND	RETRIEVE
🟩	🟩

related qualifiers

/NEW_VERSION

description

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

By default, FCX will not expand a file with an identical file name to one that already exists. An error message is generated. Using the /REPLACE qualifier is one way to override this default condition. /NEW_VERSION is another. (See the table in the description of /NEW_VERSION.)

example

```
$ FCX EXPAND DOC,MEM/REPLACE (1)
```

(1) The compressed files contained in the transport files `DOC.FCX` and `MEM.FCX` in the current default directory will be expanded. If existing files with the same version number exist for the files contained in `DOC.FCX`, the compressed files will not be expanded. Error messages will be generated. If existing files with the same version number exist for the files contained in `MEM.FCX`, the compressed files will be expanded, replacing the existing files with the same version number. All expanded files will reside in the current default directory and have the same file name and type as the corresponding compressed files.

/SELECT

Selects files from a transport file for processing.

type

Positional qualifier.

format

/SELECT=select-file-spec

operations

EXPAND	INSERT	LIST
🟩	🟩	🟩

related qualifiers

[/EXCLUDE]

description

This qualifier is used to selectively expand or list files contained in a transport file. There are no defaults for the file name or file type. The device and directory will default to the device and directory of the compressed files. If a version number is not included, it will default to * (all versions, e.g., `/SELECT=*.COM;*` and `/SELECT=*.COM` are equivalent). Relative version numbers are not supported.

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.

examples

```
$ FCX EXPAND TEST/SELECT=*.COM (1)
```

(1) The compressed files contained in the transport file `TEST.FCX` which have file type `.COM` will be expanded. All files will reside in the current default directory. The expanded files will have file names, types and version numbers which match those of the corresponding compressed files.

```
$ FCX LIST DOC/SELECT=(*.RNO,*.MEM) (2)
```

(2) The compressed files contained in the transport file `DOC.FCX` which have file type `.RNO` (all version numbers) and file type `.MEM` (all version numbers) will be listed in the brief LIST display.

/SELF_EXPAND

Requests the COMPRESS operation to create a transport file which does not require FCX to be expanded

type

Global qualifier.

format

/SELF_EXPAND=system
/SELF_EXPAND

qualifier value system

al	Alpha Linux
au	Alpha TRU64
av	Alpha VMS
iu	Itanium HP-UX
il	Itanium Linux
iv	Itanium VMS
iw	Itanium Windows
ss	Sparc Solaris
vv	VAX VMS
xl	X86 Linux
xs	X86 Solaris
xw	X86 Windows

The system indicates which is the target system for the transport file, The default is the system on which FCX is currently running.

operations

related qualifiers

/OUTPUT

description

/SELF_EXPAND requests the COMPRESS operation to create a transport file which does not require FCX to be expanded. The default is to create a file which will self-expand on the same type of system as it was created.

examples

```
$ FCX COMPRESS/SELF_EXPAND TEST.COM (1)
```

(1) The file `TEST.COM` in the current default directory (highest version only) will be compressed. The resultant self-expanding transport file will be `TEST.FCX`. This file will self-expand on the same type of system as it was created. It will have a relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS/SELF_EXPAND=VV TEST.COM (2)
```

(2) The file `TEST.COM` in the current default directory (highest version only) will be compressed. The resultant self-expanding transport file will be `TEST.FCX`. This file will self-expand on VAX systems only. It will have a relative version number of 0 and will reside in the current default directory.

/SINCE

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

type

Positional qualifier.

format

/SINCE (default is TODAY)
/SINCE=time

qualifier value time

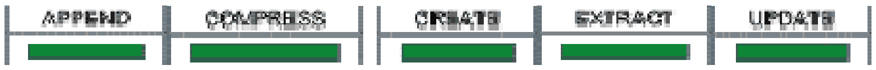
The time entered may be absolute time or delta time. Refer to the "OpenVMS DCL Concepts Manual" for the proper format. The time may also be selected using one of the following keywords.

BACKUP	The backup file time in the file's header

TODAY	The current day at 00:00:00.0 o'clock
TOMORROW	24 hours after 00:00:00.0 o'clock of the current day
YESTERDAY	24 hours before 00:00:00.0 o'clock of the current day

If no time value is specified, TODAY is assumed.

operations



related qualifiers

/BACKUP
/BEFORE
/CREATED
/EXPIRED
/MODIFIED

description

Files are selected from the input-file-spec which are dated later 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, /CREATED is assumed.

examples

```
$ FCX COMPRESS/SINCE=TODAY/MODIFIED *.FOR;* (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 for compression which have a revision time in the file header which is later than 00:00 o'clock of the current day. The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, a relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS *.C/SINCE=YESTERDAY/MODIFIED,*.FOR/SINCE=TODAY (2)
```

(2) Files in the current default directory with file type `.C` and file type `.FOR` will be candidates for compression. The files with file type `.C` will be selected based on the revision time in the file header. If the revision time is later than 00:00 o'clock of the previous day, they will be compressed. The files with file type `.FOR` will also be selected based on their creation time (CREATED is the default in this case). If the creation time is later than 00:00 o'clock of the current day, they will be compressed.

The resultant transport file will have the same file name as the first selected file; it will have a file type of `.FCX`, a relative version number of 0 and will reside in the current default directory.

```
$ FCX COMPRESS/SINCE=BACKUP/MODIFIED *.FOR;* (3)
```

(3) 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 for compression which have a revision time in the file header which is later than the backup time for the same file. The resultant transport file will have

the same file name as the first selected file; it will have a file type of `.FCX`, a relative version number of 0 and will reside in the current default directory.

/SINGLE_MODE

Requests separate transport files be generated for each input file.

type

Global qualifier.

format

/SINGLE_MODE

operations

COMPRESS

related qualifiers

/OUTPUT

description

This qualifier is used to request separate transport files be generated for each input file. The file name of each transport file will default to the input file name; the file type will default to `.FCX`. The device and directory will default to the current device and directory.

example

```
$ FCX COMPRESS/SINGLE_MODE *.COM/OUTPUT=*.XXX (1)
```

(1) All of the files with extension `.COM` in the current directory will be compressed (highest version only). A separate transport file will be generated for each of the input files. Each transport file will have the same file name as its associated input file; its extension will be `.XXX`. All FCX files will reside in the current default directory.

/STATISTICS

Request statistical information regarding the execution process be displayed.

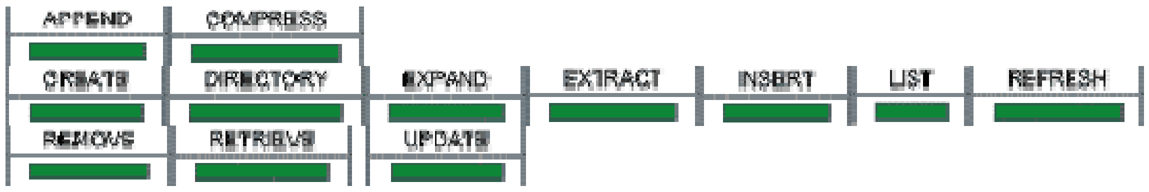
type

Global qualifier.

format

/STATISTICS
/NOSTATISTICS (default)

operations



related qualifiers

/LOG
/MONITOR

description

This qualifier causes statistical information regarding the process to be displayed to `SYS$OUTPUT` or a log-file (if /LOG is specified). 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, number of page faults, working set extent and working set quota. Other statistics are process dependent. The COMPRESS process also displays total blocks input, total compressed blocks output, and the percent reduction.

note

Block numbers in the statistics represent 512 byte blocks. If the FCX file resides on a tape, the number of compressed blocks indicated in the statistics may not match the number of blocks for the FCX file if one does a DIRECTORY of the tape.

The number of pages of virtual memory used represents the amount of dynamically allocated memory, i.e., the amount of buffers used by FCX. If this number exceeds the working set quota or the page fault rate seems excessively high, the user may wish to increase his working set extent. Increasing the extent does not necessarily increase the amount of memory available to the process nor will it automatically increase the speed of FCX.

example

```
$ FCX COMPRESS/STATISTICS SYS$SYSTEM:*.DAT;* (1)
```

(1) All files in the directory pointed to by the logical name `SYS$SYSTEM` with file type `.DAT` will be compressed. At the end of the COMPRESS operation, statistical information will be sent to `SYS$OUTPUT`. An example of this is given below.

```
                  FCX File Compression 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
   Working set extent:   2000 Working set quota: 2000
```

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

The resultant transport file will have the same file name as the first selected file; it will have a file type of .FCX, a relative version number of 0 and will reside in the current default directory.

/SUMMARY

Requests a summary display of an FCX file.

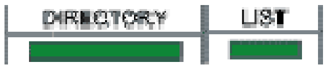
type

Global qualifier.

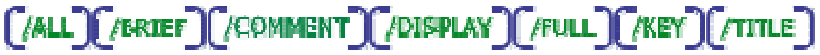
format

/SUMMARY

operations



related qualifiers



description

/SUMMARY provides for a summary display of the contents of one or more FCX files to SYS$OUTPUT. The display consists of the FCX file description (if input with /COMMENT or /TITLE during compression), the version of FCX and the operating system used during compression, and the number of compressed files contained in the FCX file. Specific information about individual compressed files is not given.

example

```
$ FCX LIST/SUMMARY SYSUAF (1)
```

(1) Sample output follows.

```
FCX File Compression Information

FCX file:              DQA0:<USER>SYSUAF.FCX;12
Creation date:         10-JUN-1993 14:42:27.48
FCX Version:           FCX Version 4.0-0
Operating system:      VMS Version V5.3
Command line:          FCX COMP SYS$SYSTEM:-SYSUAF.DAT
Created by:            USER
Default:               DQA0:<USER>
```

```
    Algorithm:            Fixed Code Dictionary
    Number of files:      1
```

/TITLE

Requests that a title be given to the discpac file being created.

type

Global qualifier

format

/TITLE=string
/TITLE="string"
/TITLE
/NOTITLE (default)

qualifier value string

The supplied string is kept as a title in the discpac file. If the string is to contain 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.

operations

CREATE

related qualifiers

/KEY

description

The supplied title is kept as part of the discpac file for easy identification of the contents of the file at a later time. It is used by the DIRECTORY operation in preparing displays of the contents of the discpac file.

examples

```
$ FCX CREATE/TITLE="com discpac" COMLIB *.COM (1)
```

(1) All files in the current directory with file type .COM (highest version only) will be compressed into the discpac COMLIB.XLB in the current directory. The contents of the title will be included in the discpac for later display by the DIRECTORY operation. The quotation marks surrounding the actual title are necessary since the title includes blanks.

```
$ FCX CREATE/TITLE=source MARLIB *.MAR (2)
```

(2) All files in the current default directory which have file type .MAR (highest version only) will be compressed into the discpac file MARLIB.XLB. The discpac will contain the contents of the title (i.e., the word "SOURCE"). Quotation marks are not necessary in this case since the title does not contain spaces or non-alphanumeric characters, however, VMS will change the characters to upper case if not enclosed in quotation marks. The title will be displayed by the DIRECTORY operation when invoked.

/TRANSLATE

Request multinational character translation.

type

Global qualifier.

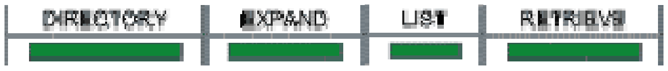
format

/TRANSLATE
/TRANSLATE=translate-table
/NOTRANSLATE (default)

qualifier value

translate-table (see below)

operations



description

This qualifier is used to specify character translation for FCX files created on a system using a different character set. Specifically, this qualifier is used during expansion of FCX files created on a PC style computer system where the default language is other than English. The default translation table is defined by a logical name **FCX_TRANS_TABLE** which points to one of the translation tables contained in the text library **SYS$SHARE:FCX_TRANS.TLB**. If the logical name is not defined, the translation-table must be specified when using this qualifier.

note

Translation of binary files may cause undesired results. This qualifier is only intended for use on ASCII files.

examples

```
$ FCX EXPAND PC_FILES/TRANSLATE (1)
```

(1) The transport file `PC_FILES` was created on a Windows system. The country code defined on the Windows system was other than US. The translation table to be used is defined by the logical name FCX_TRANS_TABLE.

The expanded files will reside in the current default directory and will have file names which match those of the compressed files. The expanded files will have version 0.

```
$ FCX EXPAND PC_FILES/TRANSLATE=FINNISH (2)
```

(2) The transport file `PC_FILES` was created on a Windows system. The country code defined on the Windows system was Finland. The translation table to be used will be the FINNISH table which resides in the text library SYS$SHARE:FCX_TRANS.TLB.

The expanded files will reside in the current default directory and will have file names which match those of the compressed files. The expanded files will have version 0.

/TRUNCATE

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

type

Positional qualifier.

format

/TRUNCATE
/NOTRUNCATE (default)

operations

EXPAND	RETRIEVE
🟩	🟩

description

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

If the compressed file is transferred to a different device, the allocation of the expanded 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 system.

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

```
$ FCX EXPAND TEST/TRUNCATE (1)
```

(1) The transport file `TEST.FCX` in the current default directory is to be expanded. The allocation of each expanded 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 expanded files will reside in the current default directory. Their file names, types and version numbers will match those of the compressed files.

/VAR_LENGTH

Requests that the transport file have variable length records.

type

Global qualifier.

format

```
/VAR_LENGTH
```

operations

COMPRESS

related qualifiers

/BLOCK_SIZE

description

By default, transport files are written as 512 byte fixed length record files. The /VAR_LENGTH qualifier provides for variable length record files with a default maximum record size of 512 bytes. If used in conjunction with the /BLOCK_SIZE qualifier, transport files may be written as variable length record files with a different maximum record size.

The purpose of this qualifier is to facilitate file transfers to different systems. Many file transfer programs can support variable length records. This allows for a shorter file transfer since the last block of the file may not be full but would be transferred anyway using fixed length records. The last block of fixed length record files is padded with zeros.

example

```
$ FCX COMPRESS/BLOCK_SIZE=510/VAR_LENGTH  *.* (1)
```

(1) All files in the current default directory will be compressed. The resultant transport file will have variable length records with a maximum record size of 510 bytes. All records in the file will

be exactly 510 bytes with the possible exception of the last record. It will contain only as many bytes as necessary to complete the transport file.

The resultant transport file will have the same file name as the first selected file; it will have a file type of .FCX, relative version number of 0 and will reside in the current default directory.

/VERIFY

Requests that files be expanded in memory with CRC checking.

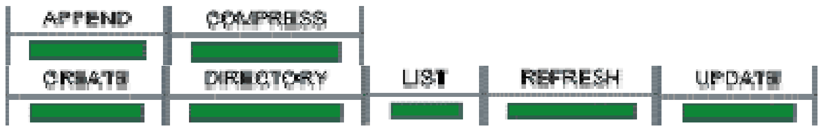
type

Global qualifier.

format

/VERIFY
/NOVERIFY (default)

operations



related qualifiers

/REBUILD
/MONITOR

description

This qualifier requests verification of the FCX file.

COMPRESS, CREATE, REFRESH, UPDATE

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

LIST, DIRECTORY

If /VERIFY is used during the LIST or DIRECTORY operation, as each compressed file is read it is expanded in memory with checking of CRCs. An error message is output if the CRCs do not match. If /MONITOR is also requested, a message indicating successful verification is output as each file is verified. Use of this qualifier will verify that the contents of the FCX file can be expanded at a later time.

This qualifier is useful for verification of an FCX file after a file transfer has taken place.

It is strongly suggested that this qualifier be used whenever the /DELETE qualifier is used.

examples

```
$ FCX COMPRESS/DELETE/VERIFY *.COM (1)
```

(1) All files in the current directory with extension `.COM` (highest version only) will be compressed. Once the transport file has been created, the verification pass will be done. If any errors occur during this pass, no files will be deleted. If the verification pass is successful, the input files will then be deleted. The resultant transport file will have the same file name as the first selected file; it will have an extension of `.FCX` and will reside in the current directory.

```
$ FCX LIST/MONITOR/VERIFY MYDIR (2)
```

(2) A brief listing of the compressed files contained in the transport file `MYDIR.FCX` will be displayed. All files will be expanded in memory with CRC checking done. Error messages will be displayed for any files which do not verify. In addition, messages indicating success (/MONITOR) will be output for all files which verify successfully.

/VERSION

Displays the version of the specified FCX product.

type

Global qualifier.

format

```
/VERSION
```

operations

DISCPAC	TRANSPORT
🟩	🟩

description

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

example

```
$ FCX TRANSPORT/VERSION (1)
```

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

```
FCX Version 7.0-0 Evaluation Kit Level x
Copyright 1989-2005 Innovative Computer Systems, Inc.
All rights reserved.
Licensee: EVALUATION KIT
Serial Number: 000000
```

Appendix

VMS File Specification Syntax

A VMS file specification consists of a device specification, a directory path, a file name, a file extension and a version number. The file name and extension are separated by a period (.). The extension and version number are separated by a semi-colon (;). The directory path consists of a set of subdirectory names separated with the period (.) character and enclosed within the bracket ([] or <>) characters. The device specification consists of a device name followed by a colon (:). The complete file specification of the file `TEST.DAT` might be

```
DUA0:[MYDIR]TEST.DAT;1
```

Logical names and search lists may be used in file specifications. For a complete description of VMS file specifications including wild cards and sticky defaults, refer to "Guide to OpenVMS File Applications".

Directories may also be specified by relative paths to the current default directory or the master file directory (root directory). The following conventions are used by FCX:

Format	Interpretation
`[000000]`	Master File Directory or Root Directory
`[]` (or none)	Current Default Directory
`[dirname]`	Subdirectory based from Root directory
`[-]`	Parent directory of current directory
`[-.dirname]`	Subdirectory based from parent directory of current directory

Directory trees are supported by FCX. Access to directory trees is given by the following conventions:

Format	Interpretation
`[...]`	Directory tree based from current directory
`[*...]`	Directory tree based from root directory (entire disk)

`[dirname...]`	Directory tree based from directory

Windows, UNIX, and VMS File Specifications

A Windows file specification consists of a drive specifier, an optional directory path, a file name and a file extension. The file name and extension are separated by a period (.). The directory path consists of a set of subdirectory names separated with the backslash (\) character and ending with the backslash (\) character.

UNIX and LInux file paths are similar to those of Windows with the exception of the drive letter. The backslash (\) character is replaced with the forward slash (/) character.

File specifications on VMS systems are quite similar; they consist of a device name, a directory path, a file name, a file type (or extension) and a version number. The file name and file type are separated by a period (.); the file type and version number are separated by a semicolon (;). The directory path consists of a set of subdirectory names enclosed in brackets ([]) and separated by periods (.). The complete file specification may be up to 256 characters,except for ODS-5 structure disks.

The following examples illustrate the difference between Windows file names and VMS file names.

Windows Filename	UNIX/Linux Filename	VMS Filename
`TEST.DOC`	`test.doc`	`TEST.DOC;1`

Directories may also be specified by relative paths to the current default directory or the master file directory (root directory). The following conventions are used:

Windows Format	UNIX/Linux Format	VMS Format	Interpretation
`\`	`/`	`[000000]`	Master File Directory or Root Directory
.`\` (or none)	`./`	`[]`	Current Default Directory
`\dirname`	`/dirname`	`[dirname]`	Subdirectory based from Root directory

`dirname\`	`dirname/`	`[.dirname]`	Subdirectory based from current directory
`..\`	`../`	`[-]`	Parent directory of current directory
`..\dirname\`	`../dirname/`	`[-.dirname]`	Subdirectory based from parent directory of current directory

Compatibility with other Systems

FCX Version 7.0 provides compatibility with FCX on Linux, Windows and UNIX systems. Files compressed on a VMS system can be expanded on these other systems and vice versa. This is true for both transport files and discpac files.

Files that reside on Windows and UNIX systems do not have record formats as do VMS files; they exist as streams of data only. Particular applications packages may define their own 'record' format but Windows has none. FCX assigns a data type to Windows and UNIX files as they are compressed. The data type is either ASCII or Binary. ASCII files are distinguished as having characters with hex values in the range 00 through 7F only. ASCII files are further distinguished by the type of VMS file they would expand to, e.g., a file with data type ASCII-LF contains line feed but does not contain the combination carriage return line feed (CR/LF). The default expansion of this file on a VMS system would be stream-LF (STMLF).

FCX files generated on Windows or UNIX may be listed using the **LIST** or **DIRECTORY** operation. If the **/FULL** qualifier is used, the data type assigned by the originating system program is displayed. If the files are then expanded using the default data type, the following RMS attributes and formats will be given to the files:

ASCII CR/LF

The file will have variable length records with the carriage return carriage control record attribute. The CR-LF combinations, which exist on Windows and are thus part of the compressed file, are stripped. The file will have the same format as most ASCII files which are created with EDT or other VMS utilities. This file This file may also be expanded using /FORMAT=STM. In this case, the CR-LF characters are kept as part of the file data.

ASCII LF

The file will have stream line feed (STMLF) format with the carriage return carriage control record attribute.

ASCII CR

The file will have stream carriage return (STMCR) format with the carriage return carriage control record attribute.

Binary

The file will have variable length records with no record attributes set. The file data will be inserted into the VMS file with no changes. Records will be defined on 512 byte boundaries. Only the last record will vary in length.

The user may override the default file type with the use of the **/ATTRIBUTE** and **/FORMAT** qualifiers or by using the **/CONFIRM=OVERRIDE** qualifier and having FCX prompt for the proper record format and attribute.

[/ATTRIBUTE] [/CONFIRM] [/FORMAT]

Multinational Character Translation Tables

FCX is currently distributed with two sample multinational character translation tables: FINNISH7 and FINNISH8. They correspond to the Finnish language translation for 7-bit and 8-bit character sets on VMS systems. These tables may be changed as necessary or copied and then modified to create tables for other languages.

Each character translation table is kept in a file which is part of a VMS library called **FCX_TRANS.TLB**. This is a text library which is created and maintained using the VMS Librarian Utility. During installation, this library is copied to the directory pointed to by the logical name FCX_MANAGER.

The VMS Librarian Utility may be used to view the contents of the library or to extract and add new character translation tables. To view the list of tables currently in the library, issue the DCL command

```
$ LIB/LIST FCX_MANAGER:FCX_TRANS.TLB
```

The DCL commands to extract and add new character translation tables are as follows:

```
$ LIB/EXTRACT=(FINNISH8)/OUTPUT=SAMPLE.DAT  FCX_MANAGER:FCX_TRANS.TLB
```

(extract the Finnish 8-bit code table into a file called SAMPLE.DAT in the current default directory)

```
$ EDIT SAMPLE.DAT
```

(edit the table to contain codes for a different language)

```
$ RENAME SAMPLE.DAT new_language.DAT
```

(rename the table to match the new language defined)

```
$ LIB/INSERT FCX_MANAGER:FCX_TRANS.TLB new_language.DAT
```

(insert the new language table into the library)

The table may replace a given one in the library using the following command:

```
$ LIB/REPLACE FCX_MANAGER:FCX_TRANS.TLB FINNISH8.DAT
```

The tables need only contain entries for which translation is required. The format of each entry is

```
index,value
```

where index is a hex value (00-FF) which indicates which character code is to be translated. Value is a hex value (00-FF) indicating the translated value. Comments are allowed in the table; these may be as long as required. Comments require the character '!' in column one.

The contents of FINNISH8.DAT follow:

```
FINNISH8.DAT

! FCX CHARACTER CODE TRANSLATION TABLE - FINNISH8.DAT
! Translates PC character codes to 8 bit VAX codes when expanding FCX
!   files on a VAX which were created by Thoro'Disc on a PC.
!
!      format of table is as follows:
!
!    PC code, VAX 8 bit code
!
! NOTES:
! 1. A comma is the only allowable delimiter between codes.
! 2. There must be only one code translation per line.
! 3. The codes must be in hex.
! 4. Comment lines are allowed but must begin with
!    a ! in column 1.
!
8F,C5
8E,C4
99,D6
86,E5
84,E4
94,F6
```