FASTCopy Reference Guide

For Windows 9x/NT/2000/XP UNIX and OpenVMS

Updated for Software Version 2.6.6

February 2, 2004

Document Version 1.0a

P.O. Box 7084, 18 Magshimim Street, Petach Tikva 49170, Israel
Tel: 972-3-9212325, Fax: 972-3-9212317,
E-mail: info@softlink.com, support@softlink.com

February 2, 2004

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)

The information in this manual has been compiled with care, but SOFTLINK makes no warranties as to accurateness or completeness, as the software described herein may be changed or enhanced from time to time. This information does not constitute commitments or representations by SOFTLINK, and is subject to change without notice. The software described in this document is furnished under license and may be used or copied only in accordance with the terms of this license.

Please direct correspondence or inquiries to:

SoftLink LTD.
18 Hamagshimim St.
P.O. Box 7084
Petach Tikva 49170
Israel

Phone:	International: +972 - 3 - 9212325
Fax:	International: +972 - 3 - 9212317
Email:	info@softlink.com
	support@softlink.com
Web site:	http://www.softlink.com/

Document Change Record

Date	Version	Description
May 29, 2003	1.0	Initial Release
February 2, 2004	1.0a	Template modifications

Table of Contents

Preface

Purpose

This document explains the functionality and implementation of FASTCopy qualifiers.

Audience

FASTCopy users.

Related Documentation

FASTCopy User Guide
FASTCopy Administrator's Guide
Active Monitor User Guide
FASTCopy Console User Guide
FASTCopy SSL Guide

Typographical Conventions

Format	**Denotes**
Blue Underlined	Hyperlinked reference to a chapter or section heading that is not a qualifier.
`Blue Underlined, New Courier`	Hyperlinked reference to qualifier
`New Courier (11)`	Qualifiers and the values that they accept
`Italicized New Courier`	Variable qualifier values (e.g. pathnames)
Bold Arial inside a paragraph	Filenames, pathnames and node names
Bold Arial	Table headings
`New Courier (10)`	Example commands

How this Manual is Organized

This manual is organized as follows:

- Chapter 1: FASTCopy Qualifiers - An Overview categorizes FASTCopy qualifiers into different groups and describes the types of attributes that each qualifier may possess.
- Chapter 2: FASTCopy Qualifiers provides a detailed explanation of FASTCopy qualifiers.
- Appendix A: Setting Time and Time-Frame Expressions describes the correct syntax for qualifiers that accept time-frame values.
- Appendix B: Encryption Methods describes the various encryption algorithms that FASTCopy supports.
- Appendix C: Saving Batch Files in the Registry explains how to save Batch qualifiers in the Windows registry.
- Appendix D: FASTCopy Quick Start is a QuickStart Guide for FASTCopy command line operations.
- Appendix E: Starting FASTCopy Daemon with the Required SSL Certificate and Key details the procedure for encrypting the private key passphrase and storing it in the Windows Registry.

New Qualifiers in this Edition of the FASTCopy Reference Guide

FASTCopy v2.6.6 now supports the MS-CAPI standard for SSL cryptography. To this end, the following qualifiers have been added:

- `SECURITY_PROVIDER` (179)
- `CERT_STORE_NAME` (195)
- `CERT_STORE_TYPE` (196)
- `CSP` (197)
- `MS_PROVIDER_TYPE` (198)
- `KEY_CONTAINER` (199)
- `CRL` (200)
- `PIN_REQUIRED` (201)

Additionally, the `CERT` (181) qualifier now accepts the client certificate’s subject field as one of its values.

1

Qualifiers – An Overview

The properties of a FASTCopy operation are controlled and modified through command line qualifiers. Qualifiers allow you to adjust many of the default parameters in a given FASTCopy file transfer. The following chapters provide detailed descriptions and examples of these qualifiers.

Note: FASTCopy Console provides the functionality of FASTCopy qualifiers through a Graphical User Interface. See the *FASTCopy Console User Guide* for further details.

For convenience, FASTCopy's qualifiers have been classified into groups defined by their respective functions, with each group comprising a separate section. The type of qualifiers that each group includes is described below. A quick reference guide for FASTCopy qualifiers is included at the end of this chapter.

Qualifier Types

FASTCopy qualifiers are divided into groups for easier reference. These groups are explained below.

Session Qualifiers

Session qualifiers control how FASTCopy handles the establishment of a communications session between the local and remote nodes. They also control other global aspects of the operation, such as the creation of on-screen or log file session reports. Session qualifiers include those used to specify remote username and password; communications port or service name, as well as qualifiers controlling the report an operation generates.

Transfer Qualifiers

Transfer qualifiers change how and what files are transferred. These qualifiers affect how files are selected on the source node and created at the target. For example, they allow you to impose simple rules that determine under what circumstances existing files should be replaced, and which of the source files' attributes should be preserved after transfer.

Scope Qualifiers

Scope qualifiers affect how FASTCopy traverses and recreates directory trees, and how it handles symbolic links to files and directories (which may be outside the selected source specification).

Performance Qualifiers

Performance qualifiers modify FASTCopy's network performance, and can be used to speed up or slow down the actual file transfer. These qualifiers control compression ratios, bandwidth limits, and differential transfer settings.

Filter Qualifiers

Filter qualifiers allow you to prevent source files that otherwise match the source file specification from being transferred. For instance, attributes such as date or owner may be used as exclusion criteria. Filter qualifiers also allow you to exclude specific files, or files matching a given specification that can include wildcards.

Recovery Qualifiers

Recovery qualifiers control FASTCopy's recovery mechanism, including limiting the number of retries and specifying a retry interval for a given operation.

Security Qualifiers

Security qualifiers control file transfer security mechanisms, for example, partner authentication and data encryption. Their use is closely linked to FASTCopy's proxy security mechanism, described in the *FASTCopy Administrator's Guide*.

Scheduling and Batch Qualifiers

Scheduling and Batch qualifiers schedule FASTCopy operations, using a platform dependent auxiliary program. This group also includes qualifiers for interacting with the scheduling mechanism.

Execution Qualifiers

Execution qualifiers activate FASTCopy's Pre- and Post-transfer processing features.

Connection Qualifiers

Connection qualifiers activate commands before and after transfer attempts, such as may be needed to establish a transient connection.

Monitor Qualifiers

Monitor qualifiers enable the monitoring of remote and local transfer operations.

Support Qualifiers

Support qualifiers provide information that may be requested by SoftLink Support. This information is used for troubleshooting various compatibility and/or licensing issues that may arise.

Qualifier Attributes

The description of each qualifier in this document is followed by a list of that qualifier's **ATTRIBUTES.** Attributes indicate different ways in which qualifiers can be used. They are as follows:

Attribute	**Description**
BATCH	A qualifier with the *BATCH* attribute can *only* be used in FASTCopy batch operations.
LIST VALUE	A qualifier with the *LIST VALUE* attribute can accept a list of values. The list should be enclosed in quotation marks, which should be escaped with backslashes if the command is issued from a UNIX shell. Inside the quotes, the values must be separated by commas. Some qualifiers can be specified with values. If the value is optional, the syntax indicates this by enclosing the value in square brackets. Most qualifiers that accept values have default values, which FASTCopy uses when no value is specified with the qualifier. Qualifiers with the *LIST VALUE* attribute can accept a single value or a list of values.
NEGATABLE	A qualifier with the *NEGATABLE* attribute can be negated by adding the prefix "no". For example, `noerror_abort` negates the `error_abort` qualifier. Qualifier negation is usually used when you wish to return to the default during an override. For example, if you specified the `confirm` qualifier in the original command, you can override it by specifying `noconfirm` in a recovery attempt. However, in one case - the `error_abort` qualifier - the default is `error_abort` and not its negation, `noerror_abort`.
NO PARAMETERS	A qualifier with the *NO PARAMETERS* attribute can only be used in FASTCopy commands that do not include any source or destination file specification parameters. It may however accept other qualifiers. (For example, `recover` is a *NO PARAMETERS* qualifier that can be used with *OVERRIDE* qualifiers).
OVERRIDE	A qualifier with the *OVERRIDE* attribute can be used in recovery commands to override default (or previously assigned) values.

POSITIONAL	A qualifier with the *POSITIONAL* attribute can be used as a positional (as well as global) qualifier. Qualifiers with the *POSITIONAL* attribute can be used differently on different source files within a single FASTCopy operation. Using positional qualifiers is explained in Using FASTCopy Qualifiers (306).
RECOVER ONLY	A qualifier with the *RECOVER ONLY* attribute can only be used in recovery commands.

Operating Systems Supported by FASTCopy Qualifiers

FASTCopy qualifiers can be used on Windows NT, Windows 9x/2000/XP, UNIX, or OpenVMS operating systems. Most qualifiers can be used on all of the aforementioned operating systems, while some cannot. The description of each qualifier in this document is followed by a list of supported operating systems.

Support for the qualifier is determined by the operating system on which the action specified by the qualifier takes place, and not the operating system from which you issue the FASTCopy command. If you specify an unsupported qualifier, FASTCopy will disregard it when the transfer operation is carried out.

Example

Symbolic links are filenames that point to other files or directories. The `follow_links` qualifier indicates to FASTCopy that if the source file specification contains a symbolic link, it should follow the link and transfer any directory that it points to, to the target node. Since only UNIX operating systems support symbolic links, the `follow_links` qualifier is supported only if the source node is a UNIX machine.

However, a FASTCopy command using the `follow_links` qualifier can be used to transfer files from a UNIX source node regardless of whether the command is issued on a UNIX, OpenVMS, Windows NT or Windows 9x/2000/XP platform.

See the `FOLLOW_LINKS` (103) qualifier for a detailed example.

Qualifiers Quick Reference

The table below alphabetically lists all of the qualifiers documented in this manual. Each qualifier is accompanied by a short description. The **OS** column displays Operating Systems on which the qualifier is supported. These can be:

ALL	Windows NT, Windows 9x/2000/XP, UNIX, OpenVMS
W	Windows NT, Windows 9x/2000/XP
U	UNIX
V	OpenVMS
W2K	Windows 2000
WNT	Windows NT
WXP	Windows XP

Qualifier	**OS**	**Action**
ACK_INTERVAL (115)	W2K WNT WXP U	The ack_interval qualifier - available from FASTCopy V2.6.5 upwards - tells the target node how many packets it can process before it must send an acknowledgement to the source node.
ASSUME (203)	ALL	Automatically replies to queries issued during a FASTCopy operation.
AUTHENTICATE (180)	W U	Verifies whether the server's Trusted Certificate is valid and/or whether the Common_Name field of the server's Trusted Certificate matches either the hostname of the server, or the value of the PEER_COMMON_NAME qualifier.
BANDWIDTH (116)	ALL	Controls the bandwidth of FASTCopy Operations.
BASE_INTERVAL (152)	ALL	Sets the time between recovery attempts
BATCH (204)	ALL	Submits a FASTCopy operation for batch execution, thereby allowing the scheduling and automatic recovery of FASTCopy transfers.

BEFORE (137)	ALL	Selects files dated before a specified time.
CA_DIR (185)	W U	Specifies the path to a directory containing Trusted Certificate Authority (CA) certificates.
CA_FILE (186)	W U	Specifies the path and filename of the Trusted Certificate Authority's certificate.
CASE (62)	ALL	Modifies the case of source filenames after transfer.
CERT (181)	W U	Specifies the path and filename of the client's SSL certificate.
CERT_STORE_NAME (195)	W2K WNT WXP	Specifies the name of the store to which you imported your MS-CAPI client certificate.
CERT_STORE_TYPE (196)	W2K WNT WXP	Specifies the type of the store to which you imported your MS-CAPI client certificate.
CHANGED_STATUS (138)	U	Selects files according to their Status Change date.
CIPHER (182)	W U	The cipher qualifier encrypts the SSL session using the selected algorithm.
CODE (167)	ALL	Specifies a mutually agreed-upon password with the command.
COMMENT (275)	U	Specifies a string of text that will be sent by a monitored FASTCopy process to the monitoring node, along with other messages the process generates.
COMPRESSED_SUFFIX (125)	ALL	Specifies file types that FASTCopy should not compress.
COMPRESSION (120)	ALL	Compresses files during transfer.
CONFIRM (63)	ALL	Instructs FASTCopy to request confirmation before performing a transfer.
CONTEXT_FILE (22)	ALL	Specifies a recovery context file, thereby allowing manual recovery from point of failure.

CONTEXT_FILE_PREFIX (25)	ALL	Modifies the location and/or name of a FASTCopy operation's context file.
CONTROLLER_DOMAIN (206)	W	Specifies the Windows NT/2000 domain name under which the user is logged on. Used when submitting a FASTCopy batch operation.
CONTROLLER_PASSWORD (207)	W	Specifies the correct password for the user account under which the batch job is submitted.
CONTROLLER_USER (208)	W	Specifies the correct username for the user account under which the batch job is submitted.
COPY_FILES (64)	ALL	Filters the source file specification.
CREATED (139)	W V	Selects files according to their creation date.
CRL (200)	W2K WNT WXP	Specifies the pathname of a file containing certificates that have been revoked by the issuing CA.
CSP (197)	W2K WNT WXP	Specifies the Cryptographic Service Provider that you want to use.
DAILY (209)	W	Specifies the time of day for the execution of a daily batch job
DELETE (66)	ALL	Deletes transferred source files after completion of the transfer operation.
DIR_LINKS (102)	U	Specifies how links to directories should be transferred.
DIRECTORY_PREFIX (67)	ALL	Abbreviates source file specifications.
DOMAIN (27)	W	Specifies that the username and password included in the command line should be treated as belonging to the specified domain.
DUMP (28)	ALL	Lists the contents of a context file.
ENCRYPT (187)	W U	Indicates that the client wishes to establish an SSL session (not necessarily authenticated).
ERROR_ABORT (29)	ALL	Aborts a multiple-file transfer operation if an error occurs.
EVERY (210)	W	Specifies the time period that should

		elapse between successive executions of a FASTCopy batch operation.
EXCLUDE (140)	ALL	Excludes files from the transfer.
EXECUTE_ONLY (227)	ALL	Executes a local/remote command independent of file transfer.
EXIT_COMMAND (229)	W U	Executes a command on termination of a FASTCopy batch operation.
EXIT_LOG_FILE (230)	W U	Specifies a log file for exit command output.
EXIT_PARAMETER (231)	W U	Passes a parameter to the exit command.
EXIT_SYNC (233)	W U	Specifies that FASTCopy should wait for the exit command to be completed before ending the operation.
EXPORT_SCRAMBLED_KEY_PHRASE (188)	W U	The `export` qualifier writes the output of the `scramble_key_phrase` qualifier to the specified file.
FACTOR (153)	ALL	Determines the factor by which base recovery interval is extended after each recovery attempt made on a FASTCopy batch operation.
FLUSH_RATE (68)	ALL	Sets the frequency with which FASTCopy asks the operating system to write a copied target file’s associated buffers to disk.
FOLLOW_LINKS (103)	U	Directs FASTCopy to follow symbolic links to directories that are not sub directories of the source root directory when `recursion=both` or `recursion=source` is specified.
FORCE (154)	ALL	Forces FASTCopy to attempt recovery, even after a non recoverable error occurs.
FORMAT_REPORT (33)	ALL	Specifies the output format of the report qualifier.
FSUBMIT_QUALS (212)	U	Users familiar with *fsubmit* can use this qualifier to modify how *fsubmit* handles the submission of the FASTCopy batch operation
GENERATE (34)	W	Enables System Administrators to

	U	provide users with a dedicated FASTCopy password.
`GID` (142)	U	Selects files only if their owners' Group Identification (GID) matches the specified *GID*.
`GROUP` (213)	U	Identifies a specific logical group as the owner of a batch operation, for the purposes of administration, security and monitoring.
`HOLD_UNTIL` (214)	ALL	Instructs FASTCopy's scheduler to hold the execution of the operation until the specified time.
`IGNORE_CWD_ERROR` (215)	W2K WXP WNT U	The `ignore_cwd_error` qualifier prevents FASTCopy from changing the current working directory to the one written in its context file.
`INFORMATION` (283)	ALL	Verifies the FASTCopy version installed on your machine.
`INIT_COMMAND` (234)	W U	Specifies a command to be executed locally before the FASTCopy operation begins.
`INIT_LOG_FILE` (235)	W U	Specifies an output file to which the output of the `init_command` qualifier should be redirected.
`INIT_PARAMETER` (236)	W U	Passes a parameter to the command specified by the `init_command` qualifier after it is activated.
`INIT_SYNC` (237)	W U	Specifies that FASTCopy should wait for the command specified with the `init_command` qualifier to be completed before starting the operation.
`INTERVAL` (126)	ALL	Specifies the interval in seconds that FASTCopy should wait before sending the next buffer.
`JOB_NAME` (216)	V	Submits a FASTCopy operation for batch execution under a particular name.
`KEEP_ALIVE_TIMER` (36)	ALL	Specifies how long FASTCopy will wait for a response from the peer node before it begins sending “keep alive” messages over the network.

KEY (190)	W U	Specifies the path and filename of the client's private key file.
KEY_CONTAINER (199)	W2K WNT WXP	Changes the default key container used by your Cryptographic Service Provider.
KEY_PHRASE (191)	W U	Specifies the passphrase for the client's private key file.
LINE_CIPHER (168)	ALL	Determines what encryption method should be used to encrypt files transferred during an operation.
LINE_ENCRYPT (170)	ALL	Encrypts one or more of the files transferred in the operation, using the key specified by the `line_phrase` qualifier and the encryption method specified with the line cipher qualifier.
LINE_PHRASE (172)	ALL	Specifies a key string to be used when encrypting files with the encryption method specified by the line cipher qualifier.
LINK_DOWN_COMMAND (270)	W U	Specifies a command that will be carried out locally in the event that a FASTCopy batch operation fails.
LINK_DOWN_PARAMETER (271)	W U	Transfers the specified parameter to a locally executed command initiated by the `link_down_command` qualifier before the FASTCopy batch operation enters a waiting phase.
LINK_UP_COMMAND (272)	W U	Specifies a command that will be carried out locally before the beginning of a FASTCopy batch operation and before every recovery attempt made on that operation.
LINK_UP_PARAMETER (273)	W U	Transfers the specified *parameter* to a locally executed command specified with the `link_up_command` qualifier.
LINKS (108)	U	Determines how FASTCopy will follow a symbolic link.
LIST (69)	ALL	Instructs FASTCopy to obtain the operation's source file specifications from a text file rather than the command line.
LIST_PREFIX (71)	ALL	Denotes a path to be prefixed to the

		source filenames in the list file (i.e. the *filename* specified by the list qualifier).
LIST_RECURSION_BASE (73)	ALL	Specifies the base path of a list of files which should not be created on the remote target.
LOCAL_COMMAND (238)	ALL	Executes a command on the local node (i.e. the node that issues the FASTCopy command), following the successful completion of the file transfer.
LOCAL_LOG_FILE (240)	W U	Specifies an output file to which the output of the local_command should be redirected.
LOCAL_PARAM (241)	ALL	Passes a parameter to the *command* specified by the local_command qualifier, after it is activated.
LOCAL_PRE_COMMAND (246)	ALL	Triggers a command on the local node (the node that issues the FASTCopy command), before the start of the file transfer operation.
LOCAL_PRE_LOG_FILE (247)	W U	Specifies an output file to which the output of the local pre-transfer command should be redirected.
LOCAL_PRE_PARAM (248)	ALL	Passes a parameter to the command specified by the local_pre_command qualifier, after it is activated.
LOCAL_PRE_QUEUE (249)	V	Submits a FASTCopy local pre-transfer command for execution under the named *queue* rather than the default OpenVMS batch queue.
LOCAL_PRE_SYNC (250)	W U	Specifies that FASTCopy should wait for a local command to be completed before beginning the transfer stage.
LOCAL_QUEUE (244)	V	Submits a FASTCopy local command for execution under the named *queue* rather than the default OpenVMS batch queue.
LOCK_SOURCE (75)	ALL	Prevents file transfer if files matching the source file specification(s) are still being written.
LOCAL_SYNC (245)	W	Specifies that FASTCopy should wait for a local command to be completed

	U	before ending the operation.
LOG_FILE (37)	ALL	Creates an output log file containing FASTCopy performance statistics.
LOG_FILE_OPEN_MODE (38)	ALL	Enables multiple FASTCopy operations to appear consecutively in the same log file or overwriting of old log file data with data generated by the current FASTCopy operation.
LOG_FORMAT (39)	ALL	Specifies the format of the log file report which FASTCopy will generate for a given operation.
LOG_NAME (44)	V	Specifies the name of the log file for a FASTCopy operation submitted for batch execution.
MAC (174)	ALL	Specifies that all messages passed between the local and the remote node should be authenticated using the given key *phrase.*
MAX_INTERVAL (158)	U V	Determines the maximum size in minutes of the interval between recovery attempts of an aborted FASTCopy operation running in batch mode.
MAX_SMALL_FILE (127)	ALL	Specifies the maximum size in bytes of files that FASTCopy will consider “small “. Files designated as “small” by this qualifier are transferred by FASTCopy using an abbreviated protocol which is more efficient than the standard FASTCopy protocol.
MIN_MODEL_SIZE (129)	ALL	Specifies the minimum file size for which the model option is activated.
MAX_WINDOW_SIZE	W2K WNT WXP	The max_window_size qualifier - available from FASTCopy V2.6.5 upwards - tells the source node how many packets it can send without receiving an acknowledgement from the target node.
MODEL (130)	ALL	Specifies a file on the target node with contents close to the source file that you are transferring. FASTCopy dynamically compares the source file to the model file during the transfer. When FASTCopy finds identical data

		in these files, instead of transferring the data from the source over the network, it takes the data from the model file (locally on the target system) to the target file.
MODEL_TABLE_SIZE	W2K WNT WXP	The model_table_size qualifier - supported from FASTCopy V2.6.5 upwards - allows you to configure the number of packets that will be compared in each model pass.
MODIFIED (143)	ALL	Selects files according to the dates that they were last modified.
MONITOR (276)	U	Requests the FASTCopy operation to report its log messages and progress to *flogicd* batch daemon, which provides centralized monitoring and control over the FASTCopy operation.
MON_LEVEL (277)	W U	Specifies the level of detail in which a monitoring process will report to the central monitoring node about the progress of the specified FASTCopy operation.
MON_NODE (279)	U	Specifies a central monitoring node (or nodes) that will be updated about the progress of the current FASTCopy operation.
MON_TYPE (280)	U	Specifies an identifying label for all the messages that the current FASTCopy process sends to a central monitoring node.
MS_PROVIDER_TYPE (198)	W2K WNT WXP	Specifies your Cryptographic Service Provider's “type”.
NEW_MONITOR (281)	W	Instructs FASTCopy to send monitoring messages to the Softlink Monitor.
NIC (45)	W2K WNT WXP	The nic qualifier forces FASTCopy to use a specific network interface (on machines with multiple interfaces).
ONCE_TIME (217)	W	Specifies a date and time for the once-only execution of a FASTCopy batch operation.

OWNER_GROUP (144)	U	Instructs FASTCopy to transfer only those files that match the *group name*.
OWNER_USER (145)	U	Instructs FASTCopy to transfer only those files that match the *username*.
PARSE_SPEC (76)	ALL	Instructs FASTCopy to replace variables in the target file specification with the actual values.
PASSWORD (46)	ALL	Specifies the password required to log into the remote node.
PATH_CREATE (78)	ALL	Creates a path on the target node according to the destination file specification.
PEER_COMMON_NAME (183)	W U	Indicates the expected contents of the server certificate's Common_Name field, when it is not the server's hostname.
PIN_REQUIRED (201)	W2K WNT WXP	Must be used if you are using a smart card or hardware token that requires the owner to interactively enter a PIN number.
PORT (47)	ALL	Changes the TCP port that FASTCopy uses to connect to the remote node.
POST_INDEPENDANT_COMMANDS (251)	W U	Specifies that if the operation is aborted after only one of the commands was triggered successfully, only the remaining command should be carried out during recovery.
POST_TRANSFER_ORDER (252)	W U	In an operation that includes both local and remote post transfer commands, the `post_transfer_order` qualifier specifies in what order these commands should be triggered.
PRE_INDEPENDANT_COMMANDS (253)	W U	In an operation that includes both local and remote pre transfer commands, the `pre_independent_commands` qualifier specifies that if the operation is aborted after only one of the commands was triggered successfully, only the remaining command should be carried out

		during recovery.
PRE_TRANSFER_ORDER (254)	W U	In an operation that includes both local and remote post transfer commands, the `pre_transfer_order` qualifier specifies in what order these commands should be triggered.
PRESERVE (80)	ALL	Specifies which of a file's attributes should be preserved when it is transferred and which should not.
PRIORITY (146)	W2K WNT WXP	The `priority` qualifier allows you to submit FASTCopy jobs according to their order of importance.
PROBE (85)	W U	Allows users to simulate and thereby verify whether a particular job will succeed.
RECOVER (159)	ALL	Recovers a previously interrupted FASTCopy operation.
RECURSION (111)	ALL	Instructs FASTCopy how to search for and copy all matching files throughout all the sub directories in the source file specification's base directory.
REMOTE_COMMAND (255)	ALL	Executes a command on the remote node, following successful completion of a file transfer.
REMOTE_LOG_FILE (257)	W U	Specifies an output file to which the output of the remote command should be redirected.
REMOTE_PARAM (258)	ALL	Passes a parameter to the remote command, after it is activated.
REMOTE_PRE_COMMAND (263)	ALL	Issues a command on the remote node, before starting a file transfer operation.
REMOTE_PRE_LOG_FILE (264)	W U	Specifies an output file to which the output of the `remote_pre_transfer` command is redirected
REMOTE_PRE_PARAM (265)	ALL	Passes a parameter to the `remote_pre_transfer` command, after it is activated.
REMOTE_PRE_QUEUE (266)	V	Submits a FASTCopy remote pre transfer command for execution under

		the named *queue* rather than the default OpenVMS batch queue.
REMOTE_PRE_SYNC (267)	W U	Specifies that FASTCopy should wait for a remote pre transfer command to be completed before starting the transfer.
REMOTE_QUEUE (261)	V	Submits a FASTCopy remote command for execution under the named *queue* rather than the default OpenVMS batch queue.
REMOTE_SYNC (262)	W U	Specifies that FASTCopy should wait for a remote command to be completed before ending the operation
REPLACE (87)	ALL	Addresses cases where the source file (or a file with the same name) already exists at the destination.
REPORT (48)	ALL	Displays\Logs the details of a FASTCopy operation.
RETRIES (164)	ALL	Specifies the maximum number of retry attempts for a FASTCopy batch operation.
RETRIEVE_REMOTE_LOG (268)	W U	Transfers a copy of the remote log to the local machine on completion of the remote command.
RMS_PRESERVE (89)	V	Preserves the unique structure of an OpenVMS file when transferring it to a different platform, such as UNIX, Windows NT or Windows 9x/2000/XP.
RMS_RESTORE (90)	V	Restores the unique structure of an OpenVMS file when retrieving it from a different platform, such as UNIX, Windows NT or Windows 9x/2000/XP.
RUN_OPTION (219)	W	Schedules the execution time frame for a batch operation.
SAVE_COMMAND (50)	ALL	Saves the FASTCopy command to the specified file instead of executing it.
SCRAMBLE_KEY_PHRASE (192)	W U	The scramble_key_phrase qualifier scrambles the passphrase to the server's private key file.
SCRAMBLED_KEY_PHRASE (193)	W	The scrambled_key_phrase

	U	qualifier provides added security by instructing FASTCopy to use the scrambled passphrase to the user's private key file instead of the actual passphrase.
`SECURITY_PROVIDER` (179)	W U	Specifies which SSL provider you want to use: OpenSSL or MS-CAPI.
`SERVICE` (52)	ALL	Changes the TCP service that FASTCopy uses to connect to the remote node.
`SILENT` (53)	ALL	Prevents the program from issuing any output message to the standard output device.
`SINCE` (147)	ALL	Selects only those files that are dated after the specified time.
`SKIP` (165)	ALL	Instructs FASTCopy to resume the transfer, beginning with the source file that follows the aborted one, during a multiple file transfer operation.
`SL_SECURITY` (176)	ALL	Specifies that if any of the files or directories in the source file specification has an associated **Security and Administration Files**, this file should be transferred to the target node along with the data file.
`TEMP_DIR` (93)	ALL	Specifies the name of a temporary target directory into which the source file will be copied under a temporary, unique name. When the transfer is complete, the file is moved to its actual destination and given its desired target name.
`TEMPLATE_FILE` (54)	ALL	Executes the command in the specified file.
`TEMPLATE_PARAMETER_FILE` (57)	ALL	Replaces any variables in the template file with the values specified in the `parameter_file`.
`TERMINATE_AFTER` (221)	ALL	Terminates a batch operation if it is not completed by the specified *time*.
`TIME_TOLERANCE` (97)	ALL	When the `copy_files` qualifier has a value of newer or `date_difference`, the `time_tolerance` qualifier specifies

		how different the time attributes of the source and destination file must be for the source file to be considered newer than or different from the target file.
TRANSLATION_TABLE (91)	ALL	The `translation_table` qualifier (trans) performs either one of the following operations: 1. Converts the character set supported on the source node, to the character set supported on the target node. -Or- 2. Converts characters within a single character set.
TRIGGER_FILE (223)	W	Specifies the name of a trigger file (*filename*) and the location (*full_path*) in which the file must be created or copied to, for the batch operation to start.
TYPE (98)	A	Determines how data is transferred between systems (binary, text, or autodetect).
UIC (149)	U	Selects files only if their owners' User Identification Code (UIC) matches the specified owner *UIC*.
UID (150)	U	Selects files only if their owners' User Identification (UID) matches the specified owner *UID*.
USER (59)	ALL	Specifies the username required to login to the remote machine.
VERIFY (99)	ALL	Controls the level of data verification performed on the file during and/or after transfer.
WARNING_ABORT (60)	ALL	Specifies that FASTCopy will abort an operation if an event that would normally generate a warning message occurs.
WARNING_STATUS (224)	ALL	Instructs FASTCopy not to perform any retries if the FASTCopy batch operation terminates with a warning.

2

FASTCopy Qualifiers

The following chapter provides a detailed description of each of the FASTCopy command line qualifiers. Each description includes:

- The name of the qualifier
- The syntax for the qualifier
- The effect of using the qualifier with the FASTCopy command
- Possible values and default values
- Attributes
- The systems on which the qualifier is supported

Session Qualifiers

This section details qualifiers that define a given FASTCopy transfer operation; These qualifiers include ones that control the program's output and log files, as well as qualifiers for establishing the context of the communications session with the remote node, such as the remote login username and password.

Session qualifiers are as follows:

- CONTEXT_FILE (22)
- CONTEXT_FILE_PREFIX (25)
- DOMAIN (27)
- DUMP (28)
- ERROR_ABORT (29)
- FORMAT_REPORT (33)
- GENERATE (34)
- KEEP_ALIVE_TIMER (36)
- LOG_FILE (37)
- LOG_FILE_OPEN_MODE (38)
- LOG_FORMAT (39)
- LOG_NAME (44)
- NIC (45)
- PASSWORD (46)
- PORT (47)
- REPORT (48)
- SAVE_COMMAND (50)
- SERVICE (52)
- SILENT (53)
- TEMPLATE_FILE (54)
- TEMPLATE_PARAMETER_FILE (57)
- USER (59)
- WARNING_ABORT (60)

CONTEXT_FILE

Syntax: `context_file=`*`file_name`*

Default*:* **fcopy_context.dat.*nnnnn***

The `context_file` qualifier specifies the name of a context file that will be created for the current FASTCopy operation. For every transfer operation that it performs, FASTCopy records its progress in a context file. This file contains complete information about the operation and is automatically deleted when the operation is successfully completed. Following a disruption, the information in the context file enables FASTCopy to continue the file transfer from the exact point of failure. When the `context_file` qualifier is used with the `RECOVER` or `DUMP` qualifiers, it specifies which context file will be read and acted upon.

The `context_file` qualifier always indicates a specific filename. This filename is used instead of the default context filename: **fcopy_context.dat.*nnnnn*** (where ***nnnnn*** is the highest version number in the current directory). On OpenVMS platforms, version numbering is supported internally, with the OpenVMS version number replacing the ***.nnnnn*** suffix. The `context_file` qualifier overrides the UNIX, Windows NT and Windows 9x/2000/XP environment variable, FCOPY_CONTEXT_FILE and the OpenVMS logical name, FASTCOPY$CONTEXT_FILE.

Note, that unlike the default context file - either **fcopy_context.dat** or whatever name was set with the UNIX, Windows NT, Windows 9x, or Windows 2000 environment variable, or with the OpenVMS logical name - any filename specified with the `context_file` qualifier is treated as a specific name. Files created using the `context_file` qualifier are done so without version numbering. Thus, if the same filename is specified as a context file for a later operation (using the `context_file` qualifier), any previous context file with the same name will be overwritten.

Since the `context_file` qualifier requires a specific filename, you must include the version number in the file specification (in the form **fcopy_context.dat.*nnnnn***) when attempting a FASTCopy `DUMP` or `RECOVER` operation (for a context file created under defaults).

The `context_file` qualifier is negatable. This means that you can specify `nocontext_file` as a qualifier. In this case, no context file will be created for the operation. This can save disk space if you are performing several small transfers, however, if `nocontext_file` is specified the operation cannot be recovered if it fails or aborts.

Examples

UNIX

```
>fcopy a.a sun:/tmp -context_file=a.context -confirm -replace
-report

############ FASTCopy started at Wed Feb 21 12:12:38 2002 ######

fcopy a.a sun:/tmp -context_file=a.context -confirm -replace -report
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD

********** FASTCopy Statistics **********

Do you want /usr/develop/yossi/fastcopy/src/a.a copied to
sun:/tmp/a.a ? [y,n,q,a]^C
>
```

At this stage the user aborts the transfer, but the context file **a.contex** exists and can be recovered from, or viewed using the `DUMP` qualifier.

UNIX

```
>fcopy -dump -context_file=a.context

FASTCopy context file contents :
Command line - fcopy a.a sun:/tmp -context_file=“a.context” -confirm
-replace -report
Local CWD - /usr/develop/yossi/fastcopy/src
Current input filename: - /usr/develop/yossi/fastcopy/src/a.a
.
.
.
```

To recover from the operation written in the context file **a.contex** you have to specify the context *filename* with the `RECOVER` qualifier.

UNIX

```
>fcopy -recover -context_file=a.context

############ FASTCopy started at Wed Feb 21 12:12:38 2002 #########

fcopy -recover -context_file=a.context
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
FCOPY-I-RECOVER - Starting recovery operation
FCOPY-I-RECOVER_TIME - Wed Jan 3 18:39:31 2002
FCOPY-I-CMD_LINE - Original command was - 'fcopy a.a sun:/tmp
-context_file=“a.context” -confirm -replace -report '

********** FASTCopy Statistics **********

Do you want /usr/develop/yossi/fastcopy/src/a.a copied to
sun:/tmp/a.a ? [y,n,q,a]y
.
.
.
FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/fastcopy/src/a.a'
FASTCopied to file 'sun:/tmp/a.a' ( 4 Bytes )
```

See also: DUMP (28) and RECOVER (159)

Attributes

NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

CONTEXT_FILE_PREFIX

Syntax: `context_file_prefix=`*directory*`|`*filename*`|`*fullspec*

The `context_file_prefix` qualifier modifies the location and/or name of a FASTCopy operation's context file. The default name of a FASTCopy context file is **fcopy_context.dat**.***nnnnn***. On Windows, .***nnnnn*** is the highest version number in the current directory. On OpenVMS platforms, version numbering is supported internally with the OpenVMS version number replacing the ***.nnnnn*** suffix.

Note: The `context_file_prefix` qualifier retains the ***.nnnnn*** suffix and therefore cannot be used in conjunction with the `context_file` qualifier, which generates a filename without the ***.nnnnn*** suffix.

The qualifier accepts any of the following values:

directory	The context file is created in the specified *directory*. ***Example*** **To create a context file in** /tmp/**:** ♦ Include the `context_file_prefix=/tmp` qualifier in your FASTCopy command line. The result is: **/tmp/fcopy_context.dat.00001**

filename	**fcopy_context.dat** is replaced with the specified *filename* and appended with the ***.nnnnn*** suffix. ***Example*** **To create a context filenamed** z.txt.*nnnnn***:** ♦ Include the `context_file_prefix=z.txt` qualifier in your FASTCopy command line.

fullspec	The context file is created in the specified directory and renamed according to the specified *filename*. The *filename* is appended with the ***.nnnnn*** suffix. ***Example*** **To create the context file in the tmp directory and rename it** z.txt: ♦ Include the `context_file_prefix=/tmp/z.txt` qualifier in your FASTCopy command line.

See also: `CONTEXT_FILE` (22)

Attributes

NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Contact support

DOMAIN

Syntax: `domain=`*`domain`*

The `domain` qualifier specifies that the username and password included in the command line should be treated as belonging to the specified *domain*. When the domain qualifier is used, the remote node checks with the domain controller whether the given user is allowed to carry out the operation.

The `domain` qualifier can only be used if the remote node is a Windows NT system.

See also: `PASSWORD` (46) and `USER` (59)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	Yes
Windows 2000	No
Windows XP	No
UNIX	No
OpenVMS	No

DUMP

Syntax: `dump`

The `dump` qualifier lists the contents of a context file. The default context file is called **fcopy_context.dat.*nnnnn*** and is located in the FASTCopy home directory, where ***nnnnn*** is the highest context file version number in the directory. On OpenVMS platforms, version numbering is supported internally, with the OpenVMS version number replacing the ***nnnnn*** suffix. You can specify a different context file by using the `context_file` qualifier with the dump qualifier. In this case, the context file's full name (including version) must be specified.

You can change the default context filename by specifying a different name with the FCOPY_CONTEXT_FILE environment variable (on UNIX, Windows NT and Windows 9x/2000/XP), or with the OpenVMS logical name: FASTCOPY$CONTEXT_FILE.

See also: `CONTEXT_FILE` (22) and `RECOVER` (159)

Attributes

NO PARAMETERS

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

ERROR_ABORT

Syntax: `error_abort`

The `error_abort` qualifier instructs FASTCopy to abort a multiple-file transfer operation if an error occurs. This is FASTCopy's default setting. It can be negated using the standard syntax negation: `noerror_abort.` When `noerror_abort` is used and an error occurs during the opening or creation of a single file, FASTCopy will issue a warning message and continue with the next file. You can override `noerror_abort` during a recovery operation by specifying `error_abort` with the recovery command. Neither the qualifier nor its negated form can modify a single-file transfer.

See also: `WARNING_ABORT` (60)

Examples

The files **a.a**, **b.b** and **c.c** are in the **c:\temp** directory. File **b.b** is currently locked by another application and therefore, cannot be transferred.

WINDOWS

```
C:\Program Files\SoftLink\FASTCopy>fcopy c:\temp\a.a,b.b,c.c
\\nt13\c:\temp -report -user=charlie -password=PASSWORD -replace
```

When FASTCopy attempts to transfer the three files to a remote location FASTCopy will abort the operation when it reaches the unavailable file **b.b.**

WINDOWS

```
FCOPY-E-OPENINP, error opening b.b for input
 -FILES-W-NOFILEFND, no files matching filespec <b.b> were found

C:\Program Files\SoftLink\FASTCopy>
```

Even though the `error_abort` qualifier is not specified in the command, its efficacy is confirmed in the screen report generated by the `report` qualifier:

Windows

```
################ FASTCopy started at Wed Feb 20 11:26:48 2002 #######

fcopy "c:\\temp\\a.a" ,"b.b" ,"c.c" "\\\\nt13\\" -user="charlie" -
password=PASSWORD -replace -report


FCOPY-I-CPYRGHT, FASTCopy V2.6 Copyright (c) 2001 by SoftLink Ltd.

       **********      FASTCopy  Statistics      **********

FCOPY-I-DEMO , FASTCopy demo license

FCOPY-I-BANNER ,  FASTCopy Temporary Evaluation License


Source  Filename : c:\temp\a.a
Target Filename : \\nt13\c:\temp\
Transfer started at : Wed Feb 20 11:26:48 2002

File size : 0 bytes

Position    %Completed   Block-size   Compressed-size   Ratio
========== ============ ============ ================= ===========
========== ============ ============ ================= ===========
Total:         100        0            0                1:1.00

FCOPY-S-FILE_COPIED, file 'c:\temp\a.a' FASTCopied to file
'\\nt13\c:\temp\a.a' ( 0 Bytes )

Transfer started at : Wed Feb 20 11:26:48 2002
Transfer ended at : Wed Feb 20 11:26:48 2002
FCOPY-E-OPENINP, error opening b.b for input
 -FILES-W-NOFILEFND, no files matching filespec <b.b> were found

C:\Program Files\SoftLink\FASTCopy>
```

The transfer aborted because FASTCopy was unable to transfer the locked file, **b.b.**

The `noerror_abort` qualifier instructs FASTCopy not to abort the operation even if one of the files (in this case, **b.b**) is found to be unavailable.

Windows

```
C:\Program Files\SoftLink\FASTCopy>fcopy c:\temp\a.a,b.b,c.c
\\nt13\c:\temp -report -user=charlie -password=PASSWORD -noerror_abort -
replace
```

The `report` qualifier generates the following report:

WINDOWS

```
################ FASTCopy started at Wed Feb 20 12:19:40 2002 #######

fcopy "c:\\temp\\a.a" ,"b.b" ,"c.c" "\\\\nt13\\c:\\temp\\" -user="charlie" -
password=PASSWORD -NOerror_abort -replace -report


FCOPY-I-CPYRGHT, FASTCopy V2.6 Copyright (c) 2001 by SoftLink Ltd.

       **********      FASTCopy  Statistics      **********

FCOPY-I-DEMO , FASTCopy demo license

FCOPY-I-BANNER ,  FASTCopy Temporary Evaluation License


Source  Filename : c:\temp\a.a
Target Filename : \\nt13\c:\temp\a.a
Transfer started at : Wed Feb 20 12:19:41 2002

File size : 0 bytes

Position    %Completed   Block-size   Compressed-size   Ratio
========== ============ ============ ================= ===========
========== ============ ============ ================= ===========
Total:         100        0            0                1:1.00

FCOPY-S-FILE_COPIED, file 'c:\temp\a.a' FASTCopied to file
'\\nt13\c:\temp\a.a' ( 0 Bytes )

Transfer started at : Wed Feb 20 12:19:41 2002
Transfer ended at : Wed Feb 20 12:19:41 2002
FCOPY-E-OPENINP, error opening b.b for input
 -FILES-W-NOFILEFND, no files matching filespec <b.b> were found

FASTCopy termination statistics report:

FCOPY-I-FILESUMM - 3 file(s) were processed by FASTCopy
        2 file has been transferred.
        0 file(s) were filtered by modification dates.
        1 file(s) failed.

FASTCopy originally started at : Wed Feb 20 12:19:40 2002
FASTCopy ended at              : Wed Feb 20 12:19:41 2002


FCOPY-W-NOTALLFILE, not all files copied by this operation

C:\Program Files\SoftLink\FASTCopy>
```

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

FORMAT_REPORT

Syntax: `format_report=full|summary|log|total`

Default: `format_report=full`

The `format_report` qualifier allows you to specify the format of the report that is generated when the `report` qualifier is used.

The `format_report` qualifier takes one of the following values:

Report Type	Description
full	This is the default. A full report contains the full source and target filenames, transfer time information and final statistics for each file. For every file, transfer statistics are listed per file buffer (packet) sent. Statistics include the packet's relative position in the file, the percentage of the file that has already been transferred, packet size before compression, packet size after compression, and the compression ratio.
summary	Reports the source and target filenames, transfer time information, size information and success and error messages.
log	Reports success messages per file and any error messages.
total	Reports the total number of successful and failed transfers among the specified files.

See also: `REPORT` (48) and `LOG_FORMAT` (39)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

GENERATE

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `generate`

The `generate` qualifier enables System Administrators to provide users with a dedicated - i.e. scrambled - FASTCopy password. Using a scrambled password, users can only perform FASTCopy operations on the remote node. This is especially useful for System Administrators who do not wish to give users full access to the remote node.

Example

Before you can generate a scrambled password, a valid user account must exist on the remote node. The following example assumes that the user's real password is "**wushu**":

To scramble a password, issue the following FASTCopy command:

WINDOWS

```
>fcopy -generate -password=wushu
```

The result is as follows:

WINDOWS

```
your scrambled password is <C2NP4NAUI1F15>
```

The user could then issue the following FASTCopy command in which `-scrambled_password=C2NP4NAUI1F15` replaces `–password=wushu`:

WINDOWS

```
fcopy SourcePath\SourceName \\RemoteName\RemPath\TargetName -repl -repo -
user=UserName -scrambled_password=C2NP4NAUI1F15
```

See also: PASSWORD (46)

Attributes

NO PARAMETERS

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

KEEP_ALIVE_TIMER

Syntax: `keep_alive_timer=`*`seconds`*

Default: 300 seconds

The `keep_alive_timer` qualifier specifies how long FASTCopy will wait for a response from the peer node before it begins sending “keep alive” messages over the network. The default value (when the qualifier is not used) is 300 seconds (5 minutes). The “keep alive” messages do not interfere with the FASTCopy operation, but they can alert the network of a failed connection. This will cause the connection to close and the operation will then fail and begin recovery rather than continue waiting.

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LOG_FILE

Syntax: `log_file=`*`[path]filename`*

Default: `log_file=FASTCopy installation directory`

The `log_file` qualifier creates an output log file containing FASTCopy performance statistics. The log file is created in the FASTCopy installation directory unless otherwise specified.

When you use the `log_file` qualifier, FASTCopy creates a file named *filename* and writes the following information to it:

- Full local and remote file specifications involved in the transfer.
- The starting time of the FASTCopy operation, and the starting time of each transfer during this operation.
- Final statistics for each file.
- The completion time of each file transfer, as well as the completion time of the entire FASTCopy operation.

For each file buffer sent, FASTCopy lists its relative position in the file, the percentage of the file that has already been transferred, file size before compression, file size after compression, and the compression ratio.

The output written to the log file is the same as the screen output generated by the report qualifier. You can modify the level of detail in a log file with the `log_format` qualifier.

See also: `REPORT` (48), `LOG_FORMAT` (39) and `FORMAT_REPORT` (33)

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LOG_FILE_OPEN_MODE

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `log_file_open_mode=append|overwrite`

The `log_file_open_mode` qualifier is used in conjunction with the `log_file=filename` qualifier and enables:

- Multiple FASTCopy operations to appear consecutively in the same log file
- Overwriting of old log file data with data generated by the current FASTCopy operation

The qualifier takes the following two values:

Value	**Description**
`append`	Appends new data to the end of any existing data in the log file specified by `log_file=filename`.
`overwrite`	Overwrites any existing data in the log file specified by `log_file=filename`.

See also: `LOG_FILE` (37)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

LOG_FORMAT

Syntax: `log_format=full|summary|log|total`

Default: `log_format=full`

The `log_format` qualifier specifies the format of the log file report, which FASTCopy will generate for a given operation. This qualifier must be used together with the `log_file` qualifier.

The `log_format` value can be:

Value	Description
Full	This is the default. A full report contains the full source and target filenames, transfer time information and final statistics for each file. For every file, transfer statistics are listed per file buffer (packet) sent. Statistics include the packet's relative position in the file, the percentage of the file that has already been transferred, packet size before compression, packet size after compression, and the compression ratio.
Summary	Reports the source and target filenames, transfer time information, size information and success and error messages.
Log	Reports success messages per file and any error messages.
Total	Reports the total number of successful and failed transfers among the specified files.

Examples of Available Log File Formats

Depending on the value specified with the `log_format` qualifier, a FASTCopy log file can contain different levels of detail, examples of which are given below.

Full

```
####### FASTCopy started at Wed Aug 21 15:17:07 2002 ( pid = 1880 )
#######

fcopy "c:\\temp\\a.a" ,"c:\\temp\\g.g" "\\\\localhost\\c:\\temp\\a.l" -
log_format="full" -NOerror_abort -replace -log_file="c:\\temp\\full.txt"
-user="charlie" -password=PASSWORD


FCOPY-I-CPYRGHT, FASTCopy V2.6.5 Copyright (c) 2002 by SoftLink Ltd.

       **********      FASTCopy  Statistics      **********

FCOPY-I-BANNER ,  QA License


Source  File name : c:\temp\a.a
Target File name : \\localhost\c:\temp\a.l
Transfer started at : Wed Aug 21 15:17:07 2002

File size : 0 bytes

Position    %Completed   Block-size   Compressed-size   Ratio
========== ============ ============ ================= ==========
========== ============ ============ ================= ==========
Total:         100        0            0                1:1.00

FCOPY-S-FILE_COPIED, file 'c:\temp\a.a' FASTCopied to file
'\\localhost\c:\temp\a.l' ( 0             Bytes )

Transfer started at : Wed Aug 21 15:17:07 2002
Transfer ended at : Wed Aug 21 15:17:07 2002
FCOPY-E-OPENINP, error opening c:\temp\g.g for input
 -FILES-W-NOFILEFND, no files matching filespec <c:\temp\g.g> were found

FASTCopy termination statistics report:

FCOPY-I-FILESUMM - 2 file(s) were processed by FASTCopy
      1 file has been transferred.
      0 file(s) were filtered by modification dates.
      1 file(s) failed.

FASTCopy originally started at : Wed Aug 21 15:17:06 2002
FASTCopy ended at              : Wed Aug 21 15:17:07 2002


FCOPY-W-NOTALLFILE, not all files copied by this operation
```

Summary

```
FCOPY-I-BANNER ,  QA License


Source  File name : c:\temp\a.a
Target File name : \\localhost\c:\temp\a.l
Transfer started at : Wed Aug 21 15:18:52 2002

File size : 0 bytes

FCOPY-S-FILE_COPIED, file 'c:\temp\a.a' FASTCopied to file
'\\localhost\c:\temp\a.l' ( 0             Bytes )

Transfer started at : Wed Aug 21 15:18:52 2002
Transfer ended at : Wed Aug 21 15:18:52 2002
FCOPY-E-OPENINP, error opening c:\temp\g.g for input
 -FILES-W-NOFILEFND, no files matching filespec <c:\temp\g.g> were found

FASTCopy termination statistics report:

FCOPY-I-FILESUMM - 2 file(s) were processed by FASTCopy
       1 file has been transferred.
       0 file(s) were filtered by modification dates.
       1 file(s) failed.

FASTCopy originally started at : Wed Aug 21 15:18:51 2002
FASTCopy ended at              : Wed Aug 21 15:18:52 2002


FCOPY-W-NOTALLFILE, not all files copied by this operation
```

Log

```
FCOPY-I-BANNER ,  QA License

FCOPY-S-FILE_COPIED, file 'c:\temp\a.a' FASTCopied to file
'\\localhost\c:\temp\a.l' ( 0             Bytes )
FCOPY-E-OPENINP, error opening c:\temp\g.g for input
 -FILES-W-NOFILEFND, no files matching filespec <c:\temp\g.g> were found

FASTCopy termination statistics report:

FCOPY-I-FILESUMM - 2 file(s) were processed by FASTCopy
       1 file has been transferred.
       0 file(s) were filtered by modification dates.
       1 file(s) failed.

FASTCopy originally started at : Wed Aug 21 15:18:28 2002
FASTCopy ended at              : Wed Aug 21 15:18:29 2002


FCOPY-W-NOTALLFILE, not all files copied by this operation
```

Total

```
FCOPY-I-BANNER ,  QA License

FCOPY-E-OPENINP, error opening c:\temp\g.g for input
 -FILES-W-NOFILEFND, no files matching filespec <c:\temp\g.g> were found

FASTCopy termination statistics report:

FCOPY-I-FILESUMM - 2 file(s) were processed by FASTCopy
       1 file has been transferred.
       0 file(s) were filtered by modification dates.
       1 file(s) failed.

FASTCopy originally started at : Wed Aug 21 15:18:06 2002
FASTCopy ended at              : Wed Aug 21 15:18:07 2002


FCOPY-W-NOTALLFILE, not all files copied by this operation
```

See also: `LOG_FILE` (37)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LOG_NAME

Syntax: `log_name=`*`name`*

The `log_name` qualifier specifies the name of the log file for a FASTCopy operation submitted for batch execution. This qualifier can only be used on OpenVMS platforms, on which FASTCopy batch operations are implemented under the VMS batch queues.

Attributes

BATCH

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	No
OpenVMS	Yes

NIC

Note: This qualifier is supported from version 2.6.5 upwards.

Syntax: `nic=`*`interface_ip_address`*

The `nic` qualifier forces FASTCopy to use a specific network interface (on machines with multiple interfaces). The value of the `nic` qualifier is IP address of the network interface that you want FASTCopy operations to use.

Attributes

NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	Contact support

PASSWORD

Syntax: `password=`*`password`*

The `password` qualifier specifies the password required to log into the remote node.

On UNIX, specifying the username and the password qualifier overrides the defaults written in your **.netrc** file.

If you use FASTCopy interactively without specifying a password (and your UNIX **.netrc** file does not include login information to the remote node), you will be prompted for a password.

Note: When working on an OpenVMS node opposite a UNIX node, keep in mind that UNIX passwords are case-sensitive. To ensure that the password reaches the UNIX node with the letters in the correct case, enclose it in quotation marks (“password“). This protects the password from being converted to upper case on the local (OpenVMS) node.

See also: `DOMAIN` (27), `USER` (59) and

GENERATE (34).

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

PORT

Syntax: `port=`*`port`*

Default: `port=5745`

The port qualifier changes the TCP port that FASTCopy uses to connect to the remote node. By default, FASTCopy uses the port defined in the services configuration file. During installation, the **fcopy$server** service is defined with a port number of 5745. 5745 is the default port used by FASTCopy and is registered with IANA.

Unless the FASTCopy daemon (*fcopyd*) on the remote node is listening to the same port, the FASTCopy operation will fail. You can activate *fcopyd* with a different port, and then instruct FASTCopy to use that port with this qualifier.

See also: `SERVICE` (52)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

REPORT

Syntax: `report`*`[=filename]`*

Note: As of FASTCopy version 2.3.1, operation reports in different formats can be sent to the screen and to a log file. To send a report to a log file, use the `log_file` qualifier instead of specifying a filename value with `report`. The *`filename`* value is still supported so that old scripts will continue to work, but its use is now depreciated.

The `report` qualifier displays FASTCopy performance statistics and transfer information and status on the standard output device. By default, FASTCopy *does not* display a report.

The `report` qualifier displays the following information about each file transfer operation:

- Full local and remote file specifications involved in the transfer.
- The starting time of the FASTCopy operation, and the starting time of each transfer within this operation.
- Final statistics for each file.
- The completion time of each file transfer, and the completion time of the entire FASTCopy operation.

In addition, for each file buffer sent, the report lists:

- Its relative position in the file.
- Percentage of the file that has already been transferred.
- File size before compression.
- File size after compression.
- Compression ratio.

If a *`[filename]`* value is specified, FASTCopy directs the performance report to the specified file. If a *`[filename]`* value is not specified, FASTCopy sends the output to the standard output device.

See also the `FORMAT_REPORT` (33) qualifier, which can be used to modify the level of detail in a report.

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

SAVE_COMMAND

Syntax: `save_command=`*`template_file_name`*

The `save_command` qualifier saves the FASTCopy command to the specified template file (*template_file_name*) instead of executing it. You can later run the saved command by specifying the filename with the `template_file` qualifier.

In addition to the standard FASTCopy syntax, a command saved in a template can also include variables, the value of which is declared when the template is used to create a FASTCopy operation. Variables can be specified instead of source or target file specifications, the remote node name, or qualifier values. Variables are identified by a preceding '$' mark and enclosing parentheses '()'.

To type the FASTCopy command in a UNIX shell:

- Enclose the entire $(variable) structure in quotation marks and place a backslash before the '$' symbol.

This will prevent the UNIX shell from attempting to modify it.

Thus, in a UNIX shell, $(variable) would be written “\$(variable)”

Examples

The following command creates a template file called **send.fct** that can be used to issue the specified command, instead of repeating the entire command.

UNIX

```
>fcopy “\$(FILE)” remote:/tmp/incoming -user=joe -password=PASSWORD
-report -replace -save_command=send.fct
```

The **send.fct** file looks like this:

UNIX

```
>more send.fct
fcopy “\$(FILE)” “remote:/tmp/incoming” -user=“joe”
-password=“PASSWORD” -report -replace
```

For an explanation of how to run a FASTCopy command based on a template, see the description of the `template_file` qualifier.

See also: `TEMPLATE_FILE` (54) and `TEMPLATE_PARAMETER_FILE` (57).

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

SERVICE

Syntax: `service=`*`service`*

Default: `service=`*`fcopy$server`*

The `service` qualifier changes the TCP service that FASTCopy uses to connect to the remote node. By default, FASTCopy uses the service defined in the services configuration file during installation. This service is usually called **fcopy$server** and assigned a port number of 5745. You can specify a different *service*, provided that both of the connected sides use the same port.

See also: `PORT` (47)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

SILENT

Syntax: `silent`

The `silent` qualifier prevents the program from issuing any output message to the standard output device. This is useful when you are calling FASTCopy from a user application and want to prevent it from sending unwanted messages to the user.

See also: `ASSUME` (203)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

TEMPLATE_FILE

Syntax: `template_file=`*`command_file`*

When the `template_file` qualifier is used, FASTCopy will execute the command in the specified *command_file*. This qualifier can be used without specifying any additional command line parameters. You can however specify overriding global qualifiers and variable values (see below).

Template files are created with the `save_command` qualifier (which saves the command line values to a file instead of executing it). You can also create a template file with a text editor, by writing a FASTCopy command in a text file. A command specified in a template file does not pass through a command line interpreter (such as a UNIX shell) before being passed to FASTCopy, and it is therefore not necessary to use backslashes before quotation marks or wildcards.

A command written in a template file can include variables, which are marked as such by using the `$(variable)` syntax.

When executing a template that includes variables, you can specify values for them in the command line or in a parameters file, using the `template_parameter_file` qualifier. If FASTCopy is unable to find a certain variable's value – i.e. if it is not specified - it will prompt the user for that value.

To specify a variable's value in the command line or the parameters file,

- Type `variable=value` where `value` is the variable value.

Example

Below are the contents of the **send.fct** template file, created with the `save_command` qualifier:

WINDOWS

```
C:\WINDOWS> more send.fct

fcopy “\$(FILE)” “\\\\localhost\\C:\\temp\\” -nouser -nopassword
-report –replace

C:\WINDOWS>
```

In the following example, the user attempts to execute the template without specifying a value for the “\$(FILE)” variable. FASTCopy prompts the user for the variable value which, when provided, enables the transfer to continue. The `format_report` qualifier complements the original report qualifier and overrides its default value to generate a shorter report.

WINDOWS

```
C:\WINDOWS>fcopy -template_file=send.fct -format_report=log
Please enter value for parameter FILE
thatch.bmp
FCOPY-S-FILE_COPIED, file 'C:\WINDOWS\THATCH.BMP' FASTCopied to file
'\\localhost\C:\temp\THATCH.BMP ' ( 598 Bytes)

FASTCopy termination statistics report:

FCOPY-I-FILESUMM - 1 file(s) FASTCopied
1 file has been transferred.
0 file(s) had identical modification dates.
0 file(s) failed.

FASTCopy originally started at : Mon May 18 16:01:30 1998
FASTCopy ended at              : Mon May 18 16:01:32 1998

FCOPY-S-FINISHED, FASTCopy operation successfully finished

>C:\WINDOWS
```

The file could also have been specified in the command line as follows:

WINDOWS

```
C:\WINDOWS>fcopy -template_file=send.fct -format_report=log
FILE=thatch.bmp
FCOPY-S-FILE_COPIED, file 'C:\WINDOWS\THATCH.BMP' FASTCopied to file
'\\localhost\C:\temp\THATCH.BMP ' ( 598 Bytes)

FASTCopy termination statistics report:

FCOPY-I-FILESUMM - 1 file(s) FASTCopied
1 file has been transferred.
0 file(s) had identical modification dates.
0 file(s) failed.

FASTCopy originally started at : Mon May 18 16:01:30 1998
FASTCopy ended at              : Mon May 18 16:01:32 1998

FCOPY-S-FINISHED, FASTCopy operation successfully finished

>C:\WINDOWS
```

For an explanation of how to specify parameters using a file, see the `template_parameter_file` qualifier.

See also: `SAVE_COMMAND` (50) and `TEMPLATE_PARAMETER_FILE` (57)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

TEMPLATE_PARAMETER_FILE

Syntax: `template_parameter_file=`*`parameter_file`*

The `template_parameter_file` qualifier replaces any variables in the template file with the values specified in the `parameter_file`. This is useful when executing a FASTCopy command with a template file that includes variables (using the `template_file` qualifier).

A `parameter_file` should be a text file containing lines in the format: variable=value. When executing the template, FASTCopy substitutes any instance of a variable (indicated as $(VARIABLE) in the template file) with the value given in the *`parameter_file`* file. If FASTCopy is unable to find a certain variable's value in the parameter file or in the command line, it will prompt the user for that value.

Examples

Below is an example of a simple template file - **send2.fct** - created with a text editor:

WINDOWS

```
C:\WINDOWS> more send2.fct

fcopy $(FILE) \\$(REMOTE)\$(TARGET_DIR)\ -nouser -nopassword
-report -replace -format_report=log

C:\WINDOWS>
```

(Note that there must be no line break in the command line)

Below is an example of a parameters file: **params.txt**.

```
REMOTE=localhost
FILE=thatch.bmp
```

The user runs the command using the **send2.fct** template and the **params.txt** file described above. FASTCopy is not able to find a value for the $(TARGET_DIR) variable in the **params.txt** file or in the command line, and therefore prompts the user for the value:

WINDOWS

```
C:\WINDOWS>fcopy -template_file=send2.fct -template_param=params.txt
Please enter value for parameter TARGET_DIR
O:data\dotan
FCOPY-S-FILE_COPIED, file 'C:\WINDOWS\THATCH.BMP' FASTCopied to file
'\\localhost\O:\data\\dotan\THATCH.BMP ' ( 598 Bytes)

FASTCopy termination statistics report:

FCOPY-I-FILESUMM - 1 file(s) FASTCopied
1 file has been transferred.
0 file(s) had identical modification dates.
0 file(s) failed.

FASTCopy originally started at : Mon May 18 17:20:22 1998
FASTCopy ended at              : Mon May 18 17:20:25 1998

FCOPY-S-FINISHED, FASTCopy operation successfully finished

>C:\WINDOWS
```

See also: `SAVE_COMMAND` (50) and `TEMPLATE_FILE` (54).

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

USER

Syntax: `user=`*`username`*

The `user` qualifier specifies the username required to login to the remote machine. On UNIX systems, you can specify a remote user name with the user qualifier to override the defaults defined in the **.netrc** file.

The username value determines the default output or input directory on the remote node and remote-file-access privileges.

If you do not specify the user qualifier – and on UNIX, a remote username is not found in your **.netrc** file - you will be prompted for a username, with your own username suggested as the default.

See also: `PASSWORD` (46) and `DOMAIN` (27)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

WARNING_ABORT

Syntax: `warning_abort`

Default: `nowarning_abort`

The `warning_abort` qualifier is similar to the `error_abort` qualifier, but specifies that FASTCopy will abort an operation if an event that would normally generate a warning message occurs. The negated form, `nowarning_abort`, is the default.

Using `warning_abort` will cause FASTCopy to abort an operation if, for example, a network partner does not support the compression method specified.

See also: `ERROR_ABORT` (29)

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

Transfer Qualifiers

Transfer qualifiers affect the way FASTCopy transfers files, for example, how files are selected, which directories are created or used, and which file attributes are preserved.

Transfer qualifiers include:

- CASE (62)
- CONFIRM (63)
- COPY_FILES (64)
- DELETE (66)
- DIRECTORY_PREFIX (67)
- FLUSH_RATE (68)
- LIST (69)
- LIST_PREFIX (71)
- LIST_RECURSION_BASE (73)
- LOCK_SOURCE (75)
- PARSE_SPEC (76)
- PATH_CREATE (78)
- PRESERVE (80)
- PROBE (85)
- REPLACE (87)
- RMS_PRESERVE (89)
- RMS_RESTORE (90)
- TEMP_DIR (93)
- TIME_TOLERANCE (97)
- TYPE (98)
- VERIFY (99)

CASE

Syntax: `case=upper|lower|exact`

The `case` qualifier defines the case of the source filenames on the target node. FASTCopy's default behavior is explained on page (316), Case Sensitivity in Filenames and File Specifications.

Possible values for the `case` qualifier are:

Value	Description
upper	FASTCopy converts transferred filenames to all upper-case letters.
lower	FASTCopy converts transferred filenames to all lower-case letters.
exact	FASTCopy preserves the exact case in transferred filenames.

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

CONFIRM

Syntax: `confirm`

Default: FASTCopy does not request confirmation.

The `confirm` qualifier instructs FASTCopy to request confirmation before a transfer operation. The default is that FASTCopy does not request confirmation.

When FASTCopy prompts the user with the names of the files to be transferred, one of the following responses can be given:

y	yes
n	no
q	quit
a	all

```
C:\Program Files\SoftLink\FASTCopy>fcopy a.txt \\localhost\c:\temp\a.a
-user=charlie -pass=charlie -confirm
FCOPY-I-BANNER ,  FASTCopy Temporary Evaluation License

 Do you want C:\Program Files\SoftLink\FASTCopy\a.txt copied to
\\localhost\c:\temp\a.a ? [y,n,q,a]
```

See also: `ASSUME` (203) and `REPLACE` (87)

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

COPY_FILES

Syntax: `copy_files=all|newer|date_difference`

Default: `copy_files=all`

The `copy_files` qualifier filters the source file specification. For source files with names matching existing files on the target node, the `copy_files` qualifier determines whether they are selected for transfer.

Value	**Description**
all	The default value. FASTCopy selects all files matching the source file specification for transfer.
date_difference	Specifies that only those files with a different date attribute than the files on the target with the same name will be selected for transfer.
newer	Specifies that only those files with a more recent date attribute than the files on the target with the same name will be selected for transfer.

If you try to transfer source files whose names match those of files on the target, FASTCopy will issue the following prompt:

```
Do you want to overwrite <target specification> from <source
specification> ? [y[N|D],n[A],q,a[N|D]]
```

Unless the value of the replace qualifier dictates a different behavior, the "overwrite" prompt will appear each time FASTCopy detects a source file with the same name as a file on the target (during a single transfer).

See also: `REPLACE` (87) and `TIME_TOLERANCE` (97)

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

DELETE

Syntax: `delete`

The `delete` qualifier deletes the source files that were copied after completion of the transfer operation. If the transfer includes post-transfer processing, such as target file rename or transfer verification, the deletion of source files will be carried out only after these steps are successfully completed.

Attributes

POSITIONAL

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

DIRECTORY_PREFIX

Syntax: `directory_prefix=`*`pathname`*

The `directory_prefix` qualifier specifies the path to the source file specification. The specified `pathname` is appended to the source file parameters to obtain the full file specification. This qualifier lets you write file specifications concisely when working with highly nested or long directory names.

Example

In the following example, the source files: **a.a**, **b.b** and **c.c**, are all located in the **c:\temp** directory.

Windows

```
>fcopy a.a,b.b,c.c \\localhost\ -user=charlie -pass=charlie -
directory_prefix=c:\temp\
```

Attributes

POSITIONAL

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

FLUSH_RATE

Syntax: `flush_rate=none|packet|file`

Default: `flush_rate=none`

The `flush_rate` qualifier sets the frequency with which FASTCopy asks the operating system to write a copied target file's associated buffers to disk.

Value	**Description**
none	This is the default value. FASTCopy depends on the operating system to handle writing the file to the disk.
packet	FASTCopy asks the operating system to write the file buffers to disk each time a packet is successfully transferred.
file	FASTCopy asks the operating system to write the file to disk when it closes the file after it has finished copying it.

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LIST

Syntax: `list=`*`filename`*

The `list` qualifier instructs FASTCopy to obtain the operation's source file specifications from a text file rather than the command line. This qualifier requires a filename value in which each line in the file specifies a different filename.

The text file must still contain the FASTCopy source file specification, and should contain either a wildcard or a pattern of characters and wildcards, which will be used to filter the file list. There is no target equivalent for the `list` qualifier, meaning that the files can originate from different places but they will all be copied to the same target directory.

Examples

The content of **list.dat** is:

WINDOWS

```
c:\autoexec.bat
c:\config.sys
c:\data\a.b
c:\data\c.d
.
.
.
.
.
```

The following FASTCopy command copies the files in the list file:

WINDOWS

```
D:> fcopy * \\remote_node\remote_target_dir -list=list.dat
```

All four files are copied to the `remote_target_dir` directory on the `remote_node` node.

See also: `LIST_PREFIX` (71) and `LIST_RECURSION_BASE` (73)

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LIST_PREFIX

Syntax: `list_prefix=`*`path`*

The `list_prefix` qualifier denotes a path to be prefixed to the source file names in the list file (i.e. the *filename* specified by the `LIST` qualifier). This is useful when the source files come from the same root directory, but are in different subdirectories.

Example

The directory structure for **Pathname 1** and **Pathname 2** is almost the same, differing only in the final subdirectory:

Pathname 1:
c:\long\long\verylong\path\data\data_a\file_1

Pathname 2:
c:\long\long\verylong\path\data\data_b\file_3

To copy **file_1** from **data_a** and **file_3** from **data_b**, *without* using the `list_prefix` qualifier, the list file needs to appear as follows:

```
c:\long\long\verylong\path\data\data_a\file_1
c:\long\long\verylong\path\data\data_b\file_3
```

Since this is not very convenient, the `list_prefix` qualifier lets you put the constant *path* in the command line and the partial specification in the list file.

Thus, the correct command for this example is:

WINDOWS

```
> fcopy * \\remote_node\target_dir -list=list.dat
-list_prefix=c:\long\long\verylong\path\data
```

With the corresponding list file being written as follows:

WINDOWS

```
data_a\file_1
data_b\file_3
```

See also: `LIST` (69) and `LIST_RECURSION_BASE` (73)

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	yes

LIST_RECURSION_BASE

Syntax: `list_recursion_base=`*`root_path`*

The `list_recursion_base` qualifier specifies the base path of a list of files, which should not be created on the remote target. This qualifier is used to instruct FASTCopy what part of the path to cut and use on the target side. When using this qualifier, you do not need to use the `path_create` qualifier if the full target path doesn't exist on the target.

Example

In our previous example (see `list_prefix`), if the command would have been:

WINDOWS

```
C:> fcopy * \\remote_node\c:\long_data -list=list.dat
-list_prefix=c:\long\long\verylong\path\data
-list_recursion_base=c:\long\long\verylong
```

Then the following files would be created underneath remote node **`c:\long_data`**:

path\data\data_a\file_1
path\data\data_b\file_3

See also: `LIST` (69), `LIST_PREFIX` (71) and `PATH_CREATE` (78).

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LOCK_SOURCE

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax: `lock_source`

The `lock_source` qualifier prevents file transfer if files matching the source file specification(s) are still being written.

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

PARSE_SPEC

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `parse_spec`

The `parse_spec` qualifier instructs FASTCopy to replace variables in the target file specification with the actual values.

The following variables can be included in the target file specification:

Variable	Replacement Value
%n	Issuer node name
%u	Issuer username
%a	Issuer IP address
%x	Unique ID consisting of *machine+user+unique_id*.

Example

```
fcopy a.a \\target_machine\d:\data\%n_%u_received.dat -parse_spec
```

Note: If you exclude the `parse_spec` qualifier, the variables will not be replaced.

Attributes

NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Contact support

PATH_CREATE

Syntax: `path_create[=last|all]`
Default: `path_create=last`

The `path_create` qualifier creates a path on the target node according to the destination file specification. Under normal circumstances, FASTCopy aborts the operation if it fails to find the path defined in the destination file specification.

The `path_create` qualifier accepts one of these two values:

Value	Description
last	The default. Creates the last directory specified in the destination file specification, if it does not exist.
all	Creates the entire path specified in the destination file specification.

Examples

The directory **dotan** exists on the **linux** node and contains no subdirectories. When the user specifies non-existent subdirectories in the destination files specification, FASTCopy generates an error:

UNIX

```
linux:/%>fcopy a.a linux:/dotan/dirA/a.a
FCOPY-E-PARDEST, invalid destination file specification
'linux:/dotan/dirA/a.a'
-FILES-E-DIRACC, failed to access directory </dotan/dirA/>
-SYS-E-ERRMSG, No such file or directory
```

The same command - now including the `path_create` qualifier – results in the creation of the sub-directory **dirA**, to which the files are successfully copied.

UNIX

```
linux:/%>fcopy a.a linux:/dotan/dirA/a.a -path_create
FCOPY-S-FILE_COPIED - File '/a.a' FASTCopied to file
'linux:/dotan/dirA/a.a' ( 6 Bytes )
FCOPY-S-FINISHED, FASTCopy operation successfully finished
```

The command does not work, however, if multiple sub-directories are specified:

UNIX

```
linux:/%>fcopy a.a linux:/dotan/dirB/dirC/a.a -path_create
FCOPY-E-PARDEST, invalid destination file specification
'linux:/dotan/dirB/dirC/a.a'
```

The solution is to specify `path_create=all`, which creates the entire path:

UNIX

```
linux:/%>fcopy a.a linux:/dotan/dirB/dirC/a.a -path_create=all
FCOPY-S-FILE_COPIED - File '/a.a' FASTCopied to file
'linux:/dotan/dirB/dirC/a.a' ( 6 Bytes )
FCOPY-S-FINISHED, FASTCopy operation successfully finished
linux:/%>
```

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

PRESERVE

Syntax: `preserve="[modification or nomodification],[protection or noprotection],[dir_protection or nodir_protection],[ownership or noownership]"`

Default: If you do not specify the `preserve` qualifier, or if you specify it without any values, FASTCopy will assume the `protection` and `dir_protection` values.

The `preserve` qualifier specifies which of a file's attributes should be preserved when it is transferred and which should not.

The qualifier accepts one or more of the following negatable values:

Value	**Description**
modification	Preserves the file modification date. If not specified (or if you specify `nomodification`), the previous modification date is lost, and the target file's modification date is the date the file was transferred. The modification information can also be preserved in cross-platform operations.
protection	Preserves file protection. If you do not specify the `preserve` qualifier or if you use it without specifying `noprotection`, FASTCopy will preserve file protection.
noprotection	FASTCopy will create the target file with read and write privileges for the owner and read privileges for the group and others. This information can also be preserved in cross-platform operations.
dir_protection	Preserves protection on directories when transferring trees. If you do not specify the `preserve` qualifier or if you use it without specifying *nodir_protection*, FASTCopy will preserve protection on directories.
nodir_protection	Creates the directory with read, write and search/execute privileges for the owner and read and search/execute privileges for the group and others. This information can only be preserved in intra-platform operations (UNIX opposite UNIX, OpenVMS opposite OpenVMS, etc.)

Value	Description
ownership	Ensures that the target file is created on the target system with the same ownership information (uid and gid in UNIX, uic in OpenVMS) as the source file. To use this qualifier, you should be authorized to set the ownership of the target file on the target system. Note that FASTCopy does not check if the uid, gid or uic are valid on the target system.
noownership	If you do not specify `preserve=ownership`, (or if you specify *noownership*), the target file's ownership information will be the same as the username under which the FASTCopy command was issued. This information can be preserved only in intra-platform operations (UNIX opposite UNIX, OpenVMS opposite OpenVMS, etc.)

The `preserve` qualifier is useful when you want to replicate a complete environment between systems.

Examples

Below is a list of files in the directory **/home/elish/tests/model** on the **linux** node. The files will be transferred to a **Windows NT** node, first with, and afterwards without the `preserve` qualifier:

UNIX

```
linux:/home/elish/tests/model%> ls -l
total 2697
drwxrwxrwx   4 root  root      1024 Aug 1 16:01 ./
drwxr-xr-x   8 elish users     1024 Aug 7 11:20 ../
-rw-r.r.   1 root    root   23641 Jul 16 14:15 a.a
-rw-r.r.   1 root    root   23641 Aug 1 14:53 a.today
-rw-r.r.   1 root    root   23641 Jul 30 12:43 b.b
-rw-r.r.   1 root    root   23641 Aug 1 14:53 b.today
-rw-r.r.   1 root    root   2540944 Aug 1 16:02 big.file
drwxr-xr-x   4 root    root      1024 Jul 16 14:15 sub1/
drwxr-xr-x   4 root    root      1024 Jul 16 14:15 sub2/
```

Example 1:

On the **Windows NT** node, the user issues a FASTCopy command to retrieve the files listed above.

The user is not concerned with preserving the original modification dates and therefore does not use the `preserve` qualifier:

WINDOWS

```
C:\USERS\eliseva\tmp\> fcopy \\linux\\home\elish\tests\model\*.* .\
-user=root -password=PASSWORD
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\a.a'
FASTCopied to file 'C:\USERS\eliseva\tmp\a.a' ( 23641 Bytes )
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\a.today'
FASTCopied to file 'C:\USERS\eliseva\tmp\a.today' ( 23641 Bytes )
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\b.b'
FASTCopied to file 'C:\USERS\eliseva\tmp\b.b' ( 23641 Bytes )
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\b.today'
FASTCopied to file 'C:\USERS\eliseva\tmp\b.today' ( 23641 Bytes )
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\big.file'
FASTCopied to file 'C:\USERS\eliseva\tmp\big.file' ( 2540944 Bytes )
FCOPY-S-FINISHED, FASTCopy operation successfully finished
```

Below is the directory on the Windows NT node after the transfer. Note that all the files now have an identical modification time, determined by the last operation carried out on them, namely, the transfer itself:

```
Volume in drive C is unlabeled Serial number is 946F:099A
Directory of C:\USERS\eliseva\tmp\*.*
8/18/96 12:35          <DIR> .
8/18/96 12:35          <DIR> ..
8/18/96 12:35           23,641 a.a
8/18/96 12:35           23,641 a.today
8/18/96 12:35           23,641 b.b
8/18/96 12:35           23,641 b.today
8/18/96 12:36        2,540,944 big.file
8/18/96 12:36            2,720 nopres.log
   2,638,228 bytes in 6 files and 2 dirs 2,640,384 bytes allocated
   6,170,624 bytes free
```

Example 2:

In this example, the user wishes to preserve the original modification dates. To accomplish this, the user issues the FASTCopy command with `preserve=modification_date:`

WINDOWS

```
C:\USERS\eliseva\tmp\> fcopy \\linux\\home\elish\tests\model\*.* .\
-user=root -password=PASSWORD -preserve=modification_date
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\a.a '
FASTCopied to file 'C:\USERS\eliseva\tmp\a.a ' ( 23641 Bytes )
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\a.today '
FASTCopied to file 'C:\USERS\eliseva\tmp\a.today ' ( 23641 Bytes )
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\b.b '
FASTCopied to file 'C:\USERS\eliseva\tmp\b.b ' ( 23641 Bytes )
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\b.today '
FASTCopied to file 'C:\USERS\eliseva\tmp\b.today ' ( 23641 Bytes )
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\big.file '
FASTCopied to file 'C:\USERS\eliseva\tmp\big.file ' ( 2540944 Bytes )
FCOPY-S-FINISHED, FASTCopy operation successfully finished
```

Below is the directory on the Windows NT node after the transfer. The directory now reflects the last time the files were modified on the source node:

WINDOWS

```
Volume in drive C is unlabeled Serial number is 946F:099A
Directory of C:\USERS\eliseva\tmp\*.*
8/18/96 12:33              <DIR> .
7/16/96 15:15               23,641 a.a
8/01/96 15:53               23,641 a.today
7/30/96 13:43               23,641 b.b
8/01/96 15:53               23,641 b.today
8/01/96 17:02            2,540,944 big.file
8/18/96 12:31                2,748 pres.log
    2,638,256 bytes in 6 files and 2 dirs 2,640,384 bytes allocated
    6,170,624 bytes free
```

Note that the modification times on the files match their original modification times on the **linux** node. This informs a user or application that, for example, the file **b.b** is older than **b.today** but newer than **a.a**, even though all the files were transferred to the Windows NT node at the same time.

Attributes

POSITIONAL
LIST_VALUE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

PROBE

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `probe=login|scan|scan,command_type`

The `probe` qualifier allows users to simulate and thereby verify whether a particular job will succeed. With the `probe` qualifier, FASTCopy performs a simulated transfer without the actual transfer of any files.

The qualifier accepts the following two values:

Value	**Description**
login	Checks that the transfer will be permitted according to the rules in the SoftLink security files on both the local and remote nodes.
scan	Simulates the FASTCopy operation without any file transfer occurring. It reports the following: ▪ Whether or not the file transfer would have taken place ▪ How many files would have been transferred ▪ Whether the user is permitted to perform the operation with the credentials he or she provided. However, since no files are actually transferred, the probe qualifier cannot test whether or not the user has permission from the remote node’s operating system to write files to the destination. Another useful feature of `-probe=scan` is its ability to compare versions of FASTCopy. If you attempt to use a qualifier in your command line that is not supported by one of the nodes, you will receive a warning message.
(scan,*command_type*)	If there are any local or remote commands in the command line, `probe=(scan`,`command_type)` verifies whether the command or executable file exists on the source/target node. It does not attempt to run the program.

The `command_type` value indicates the type of command - for example, local (post) transfer command, remote pre-transfer command, etc. - and can be one of the following:

Command Type	Description
l_pre_command	tests local pre-transfer command
l_command	tests local (post-transfer) command
r_pre_command	tests remote pre-transfer command
r_command	tests remote (post-transfer) command
l_all_commands	tests all local commands
r_all_commands	tests all remote commands
all_commands	tests all local and remote commands

Attributes

LIST_VALUE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

REPLACE

Syntax: `replace[=always|different_modification|`

`newer_modification|never]`

Default: `replace=never`

The `replace` qualifier addresses cases where the source file (or a file with the same name) already exists at the destination.

The `replace` qualifier accepts the following values as its *`replace_option`*:

Value	**Description**
always	The target file will always overwrite an existing file with the same name.
different_modification	Replaces existing target file if the modification date of the source file and the target file are different.
newer_modification	Replaces existing target file only if the modification date of the source file is later than that of the target file.
never	Does not replace an existing target file. The user will be prompted (in interactive mode) or the job will fail (when in batch mode) if a target file already exists.

If the `replace` qualifier is not specified, the user will be prompted before any existing files are overwritten (equivalent to specifying `replace=never`). Batch operations (operations submitted for execution under the batch daemon using the `batch` qualifier) will fail if a target file already exists at the destination and the `replace` qualifier is not specified.

If the `replace` qualifier is specified without a *`replace_option`* value, existing files will always be overwritten (equivalent to specifying `replace=always`), without prompting the user.

Attributes

OVERRIDE
POSITIONAL

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

RMS_PRESERVE

Syntax: `rms_preserve[=`*`fdl`*`]`

The `rms_preserve` qualifier preserves the unique structure of an OpenVMS file when transferring it to a different platform, such as UNIX, Windows NT or Windows 9x/2000/XP. When FASTCopy transfers a file from OpenVMS to another platform and the `rms_preserve` qualifier is in effect, it creates two files on the target node: one contains the source file data and has the same name as the source file, and the other contains a description of the file's VMS-specific structure and is called *`filename`*`.fdl`. If the file is then transferred back to an OpenVMS machine and the `rms_restore` qualifier is used, FASTCopy will automatically take the *`filename`*`.fdl` file and use the information it contains to reconstitute the transferred file's correct structure on the OpenVMS machine.

The `rms_preserve` and `rms_restore` qualifiers allow you to move OpenVMS files between machines through other intermediary platforms without losing the special properties of files on OpenVMS systems.

See also: `RMS_RESTORE` (90)

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	No
OpenVMS	Yes

RMS_RESTORE

Syntax: `rms_restore[=fdl]`

The `rms_restore` qualifier allows you to restore the unique structure of an OpenVMS file when retrieving it from a different platform, such as UNIX, Windows NT or Windows 9x/2000/XP. The `rms_restore` qualifier will only work if the file was originally transferred from an OpenVMS machine using FASTCopy's `rms_preserve` qualifier. When FASTCopy transfers a file from OpenVMS to another platform and the `rms_preserve` qualifier is in effect, it creates two files on the target node: one contains the source file data and has the same name as the source file, and the other contains a description of the file's VMS-specific structure and is called *`filename`*`.fdl`. If the file is then transferred back to an OpenVMS machine and the `rms_restore` qualifier is used, FASTCopy will automatically take the *`filename`*`.fdl` file and use the information it contains to reconstitute the transferred file's correct structure on the OpenVMS machine.

The `rms_preserve` and `rms_restore` qualifiers allow you to move OpenVMS files between machines through other intermediary platforms without losing the special properties of files on OpenVMS systems.

See also: RMS_PRESERVE (89)

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	No
OpenVMS	Yes

TRANSLATION_TABLE

Syntax: `translation_table=`*`table_name.tbl`*

The `translation_table` qualifier (trans) performs either one of the following operations:

1. Converts the character set supported on the source node, to the character set supported on the target node.

   -Or-

2. Converts characters within a single character set.

For example, to transfer files in ASCII format to an IBM mainframe node, the ASCII character set must be converted to the mainframe-compatible EBCDIC character set.

The table file must be a text file containing 256 numbered lines (0-255). The line number represents the source character, while the number *in* the line defines the converted target value.

Example

In the following example, the user wants the source file's tab characters to appear as spaces on the target node. The transfer is from an ASCII compatible local node to an ASCII compatible target node.

The **null.tbl** file illustrated below represents the standard ASCII character set. Therefore, including the tran=null.tbl qualifier in the FASTCopy command line will have no affect on the source file's character set.

```
0
1
2
3
4
5
.
.
.
.
255
```

However, in the **notab.tbl** file below, line number 9 (tabs) contains the number 32 (space). Therefore, including the `tran=null.tbl` qualifier in the FASTCopy command line will convert tabs to spaces.

```
0
1
2
3
4
5
6
7
8
32
10
11
.
.
.
255
```

Attributes

NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

TEMP_DIR

Syntax: `temp_dir=`*`temporary_directory_name`*

The `temp_dir` qualifier specifies the name of a temporary target directory into which the source file will be copied under a temporary, unique name. When the transfer is complete, the file is moved to its actual destination and given its desired target name. Both the temporary target name and the actual target name appear in the report header of the file's transfer.

Interactive queries such as “confirm” or “replace” relate to the real target filename.

The `temp_dir` qualifier is useful when you do not want an application on the target node to touch a file until it is completely copied. During the transfer, any recovery attempts are performed on the temporary file in the specified temporary directory.

See also: `MODEL` (130).

Examples

In the example on the following page, FASTCopy initially copies the file **a.a** to the temporary directory **/tmp** on the **sun** remote node. Only after the transfer is completed, is the file **a.a** copied from the temporary directory to **/usr/b.b** on the **sun** remote node.

UNIX

```
/usr/develop/yossi> fcopy a.a sun:/usr/b.b -report -temp_dir=/tmp -replace –
compression

##########FASTCopy started at : Thu Jan 11 14:46:36 2002##########

fcopy a.a sun:/usr/b.b -report -temp_dir=/tmp -replace
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
****************** FASTCopy Statistics ******************
Local  Filename : /usr/develop/yossi/a.a
Remote Filename : sun:/usr/b.b
Temporary output file : sun:/tmp/AAAa28207
File size : 606224 bytes

Position    %Completed   Block-size   Compressed-size    Ratio
========== ============ ============ ================= ==========
1              10         64512          33082         1:1.95
2              21         64512          34987         1:1.84
3              31         64512          36137         1:1.79
4              42         64512          30230         1:2.13
5              53         64512          33328         1:1.94
6              63         64512          36974         1:1.74
```

```
7               74          64512          35782         1:1.80
8               85          64512          62080         1:1.04
9               95          64512          35887         1:1.80
10              100         25616          7039          1:3.64
========== ============ ============ ================= ===========
Total:          100         606224         345526        1:1.75

FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/a.a' FASTCopied to
file 'sun:/tmp/AAAa28207' ( 606224 Bytes )
FASTCopy started at : Thu Jan 11 14:46:36 2002
FASTCopy ended at : Thu Jan 11 14:46:46 2002

FCOPY-I-RENAME - Temporary output file '/tmp/AAAa28207' renamed to
'/usr/b.b'
```

The report for the above operation shows that the entire file is initially copied to the temporary file **sun:/tmp/AAAa28207** on the target node. Only at the end of transfer, is the temporary file automatically renamed to the real target file: **/usr/b.b**.

The following example demonstrates the effects of the `temp_dir` qualifier on the recovery process. The `temp_dir` qualifier ensures that the real target file is not touched until the source file has arrived in its entirety. This eliminates the possibility of a partially transferred file residing on the actual target node during transfer or following a failure. This example also includes the `verify` qualifier which compares the source and destination files after both the recovery and renaming operations have taken place.

UNIX

```
/usr/develop/yossi>fcopy a.a sun:/usr/b.b -report -temp_dir=/tmp
-replace -verify –compression

########## FASTCopy started at : Thu Jan 11 14:48:45 2002 ##########

fcopy a.a sun:/usr/b.b -report -temp_dir=/tmp -replace -verify
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD

***************** FASTCopy Statistics *****************

Local  Filename : /usr/develop/yossi/a.a
Remote Filename : sun:/usr/b.b
Temporary output file : sun:/tmp/AAAa28291

File size : 606224 bytes
Position          %Completed     Block-size  Compressed-size  Ratio
============== ================ ============ ================ =======
1                10               64512         33082          1:1.95
2                21               64512         34987          1:1.84
3                31               64512         36137          1:1.79
4                42               64512         30230          1:2.13
5                53               64512         33328          1:1.94
```

```
6                63                 64512         36974         1:1.74
7                74                 64512         35782         1:1.80
^C
/usr/develop/yossi>
```

After seven blocks the user aborts the transfer. At this point the user has a partial temporary filenamed **sun:/tmp/AAAa28291** on the target node . The actual target specification, **/usr/b.b**, has not been created yet.

The following command attempts to recover the FASTCopy operation.

UNIX

```
/usr/develop/yossi>fcopy –recover

######## FASTCopy started at : Thu Jan 11 14:48:45 2002 ##########

fcopy –recover
FCOPY-I-COPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
FCOPY-I-RECOVER - Starting recovery operation
FCOPY-I-RECOVER_TIME - Thu Jan 11 14:48:56 2002
FCOPY-I-CMD_LINE - Original command was - 'fcopy a.a sun:/usr/b.b
-report -temp_dir="/tmp" -replace -verify="sync,crc,compare"
-compression'

************* FASTCopy Statistics *************
Local Filename : /usr/develop/yossi/a.a
Remote Filename : sun:/usr/b.b
Temporary output file : sun:/tmp/AAAa28291
File size : 606224 bytes
Position    %Completed   Block-size   Compressed-size   Ratio
========== ============ ============ ================= ===========
FCOPY-I-ANALYZE_START - Starting partial target file analyze ...
FCOPY-I-ANAL_OK - First 387072 bytes of target file match source file
7               74          64512         36295          1:1.78
8               85          64512         62080          1:1.04
9               95          64512         35887          1:1.80
10              100         25616         7039           1:3.64
========== ============ ============ ================= ===========
Total:          100         606224       346039          1:1.75
```

The `recover` command performs the recovery integrity checks between the source file and the temporary output file and continues the transfer from the point of failure. After the transfer is complete, FASTCopy moves the files from the temporary directory to the desired target destination and renames the target file. Additionally, FASTCopy performs the verification that was requested at the beginning of the transfer (comparing the source file to the target file).

UNIX

```
FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/a.a' FASTCopied to
file 'sun:/tmp/AAAa28291' ( 606224 Bytes )

FASTCopy started at : Thu Jan 11 14:48:45 2002
FASTCopy ended at : Thu Jan 11 14:49:02 2002

FCOPY-I-RENAME - Temporary output file '/tmp/AAAa28291' renamed to
'/usr/b.b'

FCOPY-I-VERIFY_START - Starting verification pass ...
FCOPY-I-VERIFY_OK - Files compared successfully

FASTCopy termination statistics report:
FASTCopy originally started at : Thu Jan 11 14:48:44 2002
FASTCopy ended at              : Thu Jan 11 14:49:03 2002
```

Attributes

OVERRIDE
POSITIONAL

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

TIME_TOLERANCE

Syntax: `time_tolerance=`*`seconds`*

The `time_tolerance` qualifier is used in conjunction with the `copy_files` qualifier. When the `copy_files` qualifier has a value of *`newer`* or *`date_difference`*, the `time_tolerance` qualifier specifies how different the time attributes of the source and destination file must be for the source file to be considered newer than or different from the target file.

The value of the `time_tolerance` qualifier is the maximum difference in seconds between the source and the target files' time attributes that will be ignored when these attributes are compared. If the `copy_files` qualifier has a value of `newer`, the `time_tolerance` value can be positive or negative. A positive value is added to the target file's time attribute when it is compared to the source file's time attribute. A negative value is subtracted. Only if the modified time attribute of the target is earlier than that of the source file is the source file considered `newer` and selected for transfer.

The `time_tolerance` qualifier is useful on systems where a file's time and date aren't preserved accurately during transfer. For example, on Windows 9x/2000/XP and Windows NT, because of the inaccuracies inherent in the FAT file system, a target file might be created with a modification time one second earlier or later than the source file, even when the transfer is carried out with the `preserve=modification` qualifier.

See also: `COPY_FILES` (64) and `PRESERVE` (80)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

TYPE

Syntax: `type=text|binary|autodetect`

The `type` qualifier determines how data is transferred from one system to another. Possible values are:

Value	**Description**
binary	Treats the source and target files as streams of bytes; writes any data read from the source file to the target file without modification. This is the default setting if type is not specified.
text	Treats the source and target as text files. Ensures that when writing the target file, lines from the source file are converted into lines in the proper format for the target system.
autodetect	FASTCopy determines if the source files are text or binary data and creates the target files appropriately.

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

VERIFY

Syntax: `verify=verify_type`

-Or-

Syntax: `verify=“verify_type[,verify_type,,,]”`

The `verify` qualifier controls the level of data verification performed on the file during and/or after transfer. The data verification measures are defined by the `verify_type` value.

The `verify_type` value can be one or more of the following:

Value	Description
sync	Guarantees the synchronized writing of the output file to the disk.
crc	Performs a double CRC check on each transmitted packet. This ensures that the data which was received at the target node has not been changed - either by the network or by compression and decompression algorithms.
compare	Performs a comparison on the full source and destination files following the file transfer, to ensure that they are identical.

You can use these values in any combination, to set the verification level best suited to your needs.

Specifying verify without a value activates all of the verification measures defined above.

Examples

Verification Values	FASTCopy Action
verify=crc	Performs the double CRC check on each transferred packet.
verify=\”sync,crc\”	Ensures synchronized writing to the disk, and performs a double CRC check on each transferred packet.
verify=\”compare,crc\”	Performs a double CRC check on each transferred packet, and a comparison between the source and destination files following the completion of the transfer operation. However, FASTCopy will not perform synchronized writing to the disk during the transfer.

verify=\“compare,crc,sync\”	Implements all the verification measures available. In this case it will guarantee synchronized writing to the disk, perform a double CRC check on each transferred packet, and then another compare each source and destination file following the completion of their transfer.
verify	

Attributes

POSITIONAL
OVERRIDE
LIST_VALUE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

Scope Qualifiers

This section documents qualifiers that modify the way FASTCopy handles symbolic links to files and directories.

Scope qualifiers include:

- `DIR_LINKS` (102)
- `FOLLOW_LINKS` (103)
- `LINKS` (108)
- `RECURSION` (111)

DIR_LINKS

Syntax: `dir_links[=link|dir|ignore]`

The `dir_links` qualifier specifies how links to directories should be transferred and takes the following values:

Value	Description
link	Copies the link to the directory as a symbolic link (creating only the symbolic link on the target node).
dir	Creates a directory with the name of the source symbolic link on the target node.
ignore	Does not transfer symbolic links to directories.

When the `dir_links` qualifier is used with the recursion qualifier, its default value varies according to the recursion value. The table below describes the possible `dir_link` default values.

RECURSION Value	DIR_LINK Default
recursion=both	link
recursion=source|none	ignore

See also: `FOLLOW_LINKS` (103), `PATH_CREATE` (78), `LINKS` (108) and `RECURSION` (111).

Attributes

POSITIONAL

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

FOLLOW_LINKS

Syntax: `follow_links`

The `follow_links` qualifier directs FASTCopy to follow symbolic links to directories that are not sub directories of the source root directory when `recursion=both` or `recursion=source` are specified.

If present, directories that are pointed to by symbolic links from the source directory tree will also be searched for files that should be transferred.

If multiple links to the same directory exist, FASTCopy will make sure the directory is selected only once.

While symbolic links are used only on UNIX systems, the qualifier can be used when transferring files from UNIX systems to other platforms; on the target node, the indicated directories will be created in place of the links.

See also: `DIR_LINKS` (102), `LINKS` (108) and `RECURSION`(111).

Examples

In this example, a user on the OpenVMS node **zoltar** copies a directory from the remote UNIX node **linux** to the local node. The `follow_links` qualifier is used to create any directories pointed to by symbolic links in the source directory, on the destination node. The directories are created under the name of the pointing link.

Below is the directory **/tmp/toptree** on the UNIX node **linux**. The directory contains the file **a.exe** - and the sub directory: **subdir1**. The sub directory contains one file - **b.exe** - and one symbolic link - **NiceLink** - which points to a sub directory in the directory **/tmp/othertree**:

UNIX

```
linux:/tmp/toptree> ls -l
-rw-r.r.  1 root        root         697512 Aug 14 11:29 a.exe
subdir1:
total 690
lrwxrwxrwx  1 root        root         24 Aug 14 11:41 NiceLink ->
/tmp/othertree/othersub1/
-rw-r-r- 1 root           root         697512 Aug 14 11:29 b.exe
subdir2:
total 690
```

Below is the directory **/tmp/othertree** on the same node. It contains the file **aa.exe** and the sub directory **othersub1**. The sub directory contains a single file **bb.exe**. The sub directory **/tmp/othertree/othersub1** is the directory pointed to by the symbolic link: **/tmp/toptree/subdir1/NiceLink**:

```
linux:/tmp/othertree> ls -l
-rw-r-r-   1   root   root      697512 Aug 14 11:31 aa.exe
othersub1:
total 690
-rw-r-r-  1    root   root     697512 Aug 14 11:32 bb.exe
```

Below is the working directory on the OpenVMS node **zoltar** where the files are copied to:

OPENVMS

```
Z_ELISHEVA.dir

Directory USERS0:[ELISHEVA]

LOGIN.COM;3         MODEL.DIR;1           SRC.DIR;1
TEMP.DIR;1

Total of 4 files.

Z_ELISHEVA.
```

From the above directory on the OpenVMS node **zoltar**, the user issues the following command:

OPENVMS

```
Z_ELISHEVA.fcopy linux::[tmp.toptree]*.* [] /report /recursion=both
/follow_links /replace /format_report=summary
```

The following report is generated for the FASTCopy operation:

OPENVMS

```
########### FASTCopy started at Sun Aug 18 14:05:46 2002 ###########

fcopy “linux::[tmp.toptree]*.*” “[]” /report=““ /recursion=“both”
/follow_links /replace=“always” /format_report=“summary” /error_abort
```

```
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD

FCP-I-LICD FASTCopy demonstration license

********** FASTCopy Statistics **********

Name (linux::ELISHEVA): root
Password (linux::root):
FCOPY-I-TREE - Building directory tree list

Local Filename : USERS0:[ELISHEVA]A.EXE
Remote Filename : linux::[tmp.toptree]a.exe
Transfer started at : Sun Aug 18 14:05:53 2002

File size : 697512 bytes

FCOPY-S-FILE_COPIED - File 'linux::[tmp.toptree]a.exe ' FASTCopied to
file 'USERS0:[ELISHEVA]A.EXE;2' ( 697512 Bytes )

Transfer started at : Sun Aug 18 14:05:53 2002
Transfer ended at : Sun Aug 18 14:06:24 2002
```

The file **a.exe** is copied from the source base directory to the target base directory. Next, the sub directory **subdir1** is copied to the target base directory. The file **b.exe** is copied from the sub directory **subdir1** in the source base directory to the sub directory of the same name in the target base directory:

OpenVMS

```
Local Filename  : USERS0:[ELISHEVA.SUBDIR1]B.EXE
Remote Filename : linux::[tmp.toptree.subdir1]b.exe
Transfer started at : Sun Aug 18 14:06:25 2002

File size : 697512 bytes

FCOPY-S-FILE_COPIED - File 'linux::[tmp.toptree.subdir1]b.exe'
FASTCopied to file 'USERS0:[ELISHEVA.SUBDIR1]B.EXE;1' (697512 Bytes)

Transfer started at : Sun Aug 18 14:06:25 2002
Transfer ended at : Sun Aug 18 14:06:56 2002
```

Next, FASTCopy follows the symbolic link **NiceLink** in the source sub directory: **subdir1**. The link points to a directory, prompting FASTCopy to create a directory inside the target sub directory **subdir1** with the same name as the link and copies the contents of the linked directory (**/tmp/othertree/other1** on the **linux** source node) to that directory:

OpenVMS

```
Local Filename : USERS0:[ELISHEVA.SUBDIR1.NICELINK]BB.EXE
Remote Filename : linux::[tmp.toptree.subdir1.NiceLink]bb.exe
Transfer started at : Sun Aug 18 14:06:58 2002

File size : 697512 bytes

FCOPY-S-FILE_COPIED - File
'linux::[tmp.toptree.subdir1.NiceLink]bb.exe' FASTCopied to file
'USERS0:[ELISHEVA.SUBDIR1.NICELINK]BB.EXE;1' ( 697512 Bytes )

Transfer started at : Sun Aug 18 14:06:58 2002
Transfer ended at : Sun Aug 18 14:07:29 2002
```

OpenVMS

```
Transfer started at : Sun Aug 18 14:07:29 2002
Transfer ended at : Sun Aug 18 14:07:30 2002

FASTCopy termination statistics report:
FCOPY-S-FILESUMM - 10 files FASTCopied out of 10

FASTCopy originally started at : Sun Aug 18 14:05:46 2002
FASTCopy ended at : Sun Aug 18 14:08:36 2002
```

And finally, the directories on the OpenVMS node **zoltar** after the transfer:

OpenVMS

```
Directory USERS0:[ELISHEVA]

A.EXE;2              A.EXE;1               FCOPY_CONTEXT.DAT;1
LOGIN.COM;3          MODEL.DIR;1          SRC.DIR;1
SUBDIR1.DIR;1          TEMP.DIR;1

Total of 8 files.

Directory USERS0:[ELISHEVA.SUBDIR1]

B.EXE;1              NICELINK.DIR;1

Total of 2 files.
```

```
Directory USERS0:[ELISHEVA.SUBDIR1.NICELINK]

BB.EXE;1

Total of 1 files.
```

Attributes

POSITIONAL
NEGATABLE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

LINKS

Syntax: `links=`*`[ignore|link|file|both]`*
Default: `links=`*`file`*

The `links` qualifier determines how FASTCopy will follow a symbolic link.

The qualifier accepts the following values:

Value	FASTCopy Action
ignore	Ignores symbolic links.
link	Copies the symbolic link only, creating an identical symbolic link on the target system, but without copying the data file that the symbolic link points to.
file	Copies the file pointed by the symbolic link.
both	Copies the symbolic link and the file that it indicates to the target system.

If `links` is not specified, FASTCopy will ignore symbolic links. If `links` is specified without a value, FASTCopy will copy the file that the symbolic link points to.

See also: `FOLLOW_LINKS` (103), `DIR_LINKS` (102) and `RECURSION` (111).

Examples

The file **a.a** and a symbolic link to this file - **link_to_a.a** - are in the user's current working directory **/tmp** on node **alf**.

UNIX

```
alf:/tmp> ls -l a.a
-rwxr-xr-x 1 root other 1321286 Feb 18 09:08 a.a
alf:/tmp> ls -l link_to_a.a
lrwxrwxrwx 1 root other 3 Feb 18 13:06 link_to_a.a -> a.a
```

When the FASTCopy command to transfer **link_to_a.a** includes `links=ignore`, the file is ignored and no files are copied.

UNIX

```
alf:/tmp>fcopy link_to_a.a sun2:/tmp1/ -replace -links=ignore
FCOPY-W-NOFILES, No files copied by this operation
```

When `links=link` is included in the FASTCopy command to transfer **link_to_a.a**, only the symbolic link is copied, creating an identical symbolic link in the target directory.

UNIX

```
alf:/tmp> fcopy link_to_a.a sun2:/tmp1 -replace -links=link
FCOPY-S-LINK_COPIED - Symbolic link '/tmp/link_to_a.a->a.a'
FASTCopied to symbolic link 'sun2:/tmp1/link_to_a.a->a.a'
```

When `links=file` is included in the FASTCopy command to transfer **link_to_a.a**, the file **a.a** that the symbolic link points to is transferred to the target directory.

UNIX

```
alf:/tmp> fcopy link_to_a.a sun2:/tmp1 -replace -links=file
FCOPY-S-FILE_COPIED - File '/tmp/link_to_a.a->a.a' FASTCopied to file
'sun2:/tmp1/link_to_a.a' ( 1321286 Bytes )
```

When `links=both` is included in the FASTCopy command to transfer **link_to_a.a**, both the symbolic link and the file **a.a** that the symbolic link points to are transferred to the target directory.

UNIX

```
alf:/tmp> fcopy link_to_a.a sun2:/tmp1 -replace -links=both
FCOPY-S-FILE_COPIED - File '/tmp/link_to_a.a->a.a' FASTCopied to file
'sun2:/tmp1/link_to_a.a->a.a'  ( 1321286 Bytes )
```

In the target directory on the node **sun2**, the following files will exist after the completion of the transfer described above (links=*both*):

UNIX

```
sun2:/tmp1> ls -l link_to_a.a
lrwxrwxrwx 1 root other 3 Feb 18 13:14 link_to_a.a -> a.a
sun2:/tmp1> ls -l a.a
-rwxr-xr-x 1 root other 1321286 Feb 18 09:08 a.a
```

Attributes

POSITIONAL

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

RECURSION

Syntax: `recursion[=`*`recurs_value`*`]`

The `recursion` qualifier instructs FASTCopy how to search for and copy all matching files throughout all the sub directories in the source file specification's base directory.

The *`recurs_value`* can be `none`, `source`, or `both`.

Value	Description
none	Default value when recursion is not specified. Searches for matching files in the source file specification base directory only.
source	Searches for source files in all sub directories of the source file specification's base directory. Copies all matching files to the base directory of the destination file specification
both	Default value when recursion is specified without a value. Performs the following actions: ▪ Searches for source files throughout all the sub directories of the source file specification's base directory. ▪ Recreates the directory tree structure of the matching source files under the base directory of the destination file specification. ▪ Copies the source files from their sub directories to the corresponding sub directories of the destination file specification.

Examples

In the following example, the file **a.a** resides on the local node in the user's current working directory and in two sub directories: **sub1** and **sub2**.

UNIX

```
/test> ls -l a.a
-rw-rw-rw-    1 root   sys           48 Jan 14 13:57 a.a
/test1> ls -l sub1/a.a
-rw-rw-rw-    1 root   sys           48 Jan 14 13:57 sub1/a.a
/test1> ls -l sub2/a.a
```

```
-rw-rw-rw-    1 root   sys           48 Jan 14 13:57 sub2/a.a
/test1>
```

The following command copies all the files named **a.a** on the local node to the directory **/test_area** on the remote node **linux**.

UNIX

```
/test1> fcopy a.a linux:/test_area -recurs=source

FCOPY-I-TREE - Building directory tree list
FCOPY-S-FILE_COPIED - File '/test1/a.a' FASTCopied to file
'linux:/test_area/a.a' ( 48 Bytes )
```

The file **a.a** from the **sub1** subdirectory replaces the file **a.a** in the target directory and is in turn replaced by the file **a.a** from the **sub2** subdirectory. Since the `replace` qualifier was not specified in the original command, FASTCopy prompts the user for permission to overwrite the existing files.

UNIX

```
Do you want to overwrite linux:/test_area/a.a from /test1/sub1/a.a ?
[y,n,q,a]y
FCOPY-S-FILE_COPIED - File '/test1/sub1/a.a' FASTCopied to file
'linux:/test_area/a.a' ( 48 Bytes )
Do you want to overwrite linux:/test_area/a.a from /test1/sub2/a.a ?
[y,n,q,a]y
FCOPY-S-FILE_COPIED - File '/test1/sub2/a.a' FASTCopied to file
'linux:/test_area/a.a' ( 48 Bytes )
```

See also: `DIR_LINKS` (102) and `FOLLOW_LINKS` (103).

In the following example, FASTCopy creates the sub directories **sub1** and **sub2** under the directory **/test_area** on the remote node **linux**, and copies the corresponding **a.a** source files to them.

UNIX

```
/test1> fcopy a.a linux:/test_area -recurs=both

FCOPY-I-TREE - Building directory tree list
FCOPY-S-FILE_COPIED - File '/test1/a.a' FASTCopied to file
```

```
'linux:/test_area/a.a' ( 48 Bytes )
FCOPY-S-FILE_COPIED - File '/test1/sub1/a.a' FASTCopied to file
'linux:/test_area/sub1/a.a' ( 48 Bytes )
FCOPY-S-FILE_COPIED - File '/test1/sub2/a.a' FASTCopied to file
'linux:/test_area/sub2/a.a' ( 48 Bytes )
/test1>
```

Attributes

OVERRIDE
POSITIONAL

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

Performance Qualifiers

This section details qualifiers that modify the network performance of FASTCopy transfers.

Performance qualifiers include:

ACK_INTERVAL

Note: This qualifier is supported from version 2.6.5 upwards.

Syntax: `ack_interval=`*`numeric_value`*

The `ack_interval` qualifier - available from FASTCopy V2.6.5 upwards - tells the target node how many packets it can process before it must send an acknowledgement to the source node. In large files, this may have a negative impact on the overall transfer speed. In most cases, increasing the number of packets required before each acknowledgement will increase the overall transfer speed.

See also: `MAX_WINDOW_SIZE` (128)

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Contact support

BANDWIDTH

Syntax: `bandwidth=`*`“bandwidth_limit [time_frame] [,bandwidth [time_frame]]”`*

The `bandwidth` qualifier controls the transfer rate of a given FASTCopy operation, ensuring that the line bandwidth consumed by the operation never exceeds the specified *`bandwidth_limit`*. The `bandwidth` qualifier can also be implemented in such a way that the specified *`bandwidth_limit`* varies according to a predefined *time_frame*.

See Appendix A: Setting Time and Time-Frame Expressions (285).

The *`bandwidth_limit`* can either be **absolute** (a set value) or **relative** (a percentage of available bandwidth):

Absolute Bandwidth

This is an absolute numeric value specifying the maximum bandwidth that the transfer is permitted to consume. The value is specified as a number (which can also be a decimal fraction, such as 7.6 or 9.45) followed by two letters to denote the units used: **K** (for kilo) or **M** (for mega) followed by **B** (for bytes) or **b** (for bits):

Syntax: `nnnn.nnnn[K or M][B or b]`

Examples

- 256KB = 256 kilobytes per second
- 1.6Mb = 1.6 megabits per second

The default is bits per second. Thus:

- 256 = 256 bits per second
- 256K = 256 kilobits per second

Note: On OpenVMS platforms, the bandwidth limit value must be enclosed in quotation marks.

In the following example, the FASTCopy command specifies an absolute bandwidth limit of 25 KBytes per second.

UNIX

```
> fcopy print.dat sun:/tmp -report -comp=none -bandwidth=25KB
######## FASTCopy started at Mon Mar 4 05:55:03 2002 ##########
fcopy print.dat sun:/tmp -report=““ -compression=“none”
-bandwidth=“25KB” -error_abort
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
********** FASTCopy Statistics **********
Local Filename : /tmp/print.dat
Remote Filename : sun:/tmp/print.dat
Transfer started at : Mon Mar 4 05:55:03 2002
File size : 886222 bytes

Position   %Completed   Block-size  Compressed-size Ratio
========== ============ =========== =============== =========
1               7         64512        64512         1:1.00
.
.
.
12              87        64512        64512         1:1.00
13              94        64512        64512         1:1.00
14              100       47566        47566         1:1.00
========== ============ ============ ============== ==========
Total:          100       886222       886222        1:1.00

FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/print.dat ' FASTCopied to file
'sun:/tmp/print.dat' ( 886222 Bytes )

Transfer started at : Mon Mar 4 05:55:03 2002
Transfer ended at : Mon Mar 4 05:55:38 2002
```

The file size that was transferred was 886222 bytes, which when divided by 35 (the transfer time) gives a transfer rate of 25KBytes per second.

Relative Bandwidth

Relative bandwidth is the maximum *percentage* of available bandwidth that a transfer is permitted to consume. The relative bandwidth limit varies in relation to the line's current load. The relative *bandwidth_limit* is specified as an integer from 1-100, immediately followed by the percentage symbol (%).

Example

To specify a relative bandwidth of 22%, add the following bandwidth qualifier to the FASTCopy command:

```
bandwidth=22%
```

Varying Bandwidth Consumption According to Time Periods

The bandwidth qualifier can also be specified with a time frame expression that defines a time frame within which the *bandwidth_limit* applies.

You can specify multiple *time_frame*(s) and corresponding bandwidth_limit(s) in a single transfer operation.

Example

Joe wishes to submit a single large-scale FASTCopy operation that consumes no more than 25% of the line between 8:00-17:00, but 85% of the line between 17:00-24:00.

The bandwidth qualifier for this operation would be written as follows:

```
bandwidth=\” 25% h:08-17 , 85% h:17-24 \”
```

See also: INTERVAL (126).

Attributes

POSITIONAL
OVERRIDE
LIST VALUE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

COMPRESSION

Syntax: `compression[=none|normal|ap|power]`
Default: `compression=ap`

The `compression` qualifier uses the selected compression algorithm to compress the data transferred by FASTCopy.

The `compression` qualifier takes one of the following values:

Value	**Properties**
none	FASTCopy performs no compression on the transmitted data.
normal	FASTCopy uses an efficient compression algorithm.
ap	FASTCopy uses an efficient compression algorithm, which has better compression results than the *normal* algorithm.
power	FASTCopy uses a powerful compression algorithm, which produces better compression results on executable files than the *normal* and *ap* algorithms do. However, it consumes more CPU time.

The default *`compression_type`* (when the `compression` qualifier is not used) is none. If compression is specified without a value, the *`compression_type`* is ap.

For further information on which compression method is best suited to your needs, please refer to the *FASTCopy User's Guide.*

Examples

The next three pages provide examples of the different compression methods. In each case, a large file is being transferred to the local **Windows NT** node from the remote UNIX node **linux**. Each time, a different compression method is used.

The following transfer uses the `normal` compression algorithm:

WINDOWS

```
C:\USERS\eliseva\tmp>fcopy \\linux\\home\elish\tests\model\file.exe
.\\ -report -replace -compression=normal -user=root
-password=PASSWORD
########### FASTCopy started at Sun Aug 18 13:21:48 2002 #######

fcopy “\\\\linux\\\\home\\elish\\tests\\model\\file.exe” “.\\”
-report -replace -compression=“normal” -user=“root”
-password=PASSWORD -error_abort

FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD

FCP-I-LICD FASTCopy demonstration license

********** FASTCopy Statistics **********

Local Filename  : C:\USERS\eliseva\tmp\file.exe
Remote Filename : \\linux\home\elish\tests\model\file.exe
Transfer started at : Sun Aug 18 13:21:51 2002

File size : 3662016 bytes

Position   %Completed   Block-size   Compressed-size   Ratio
========== ============ ============ ===============   ========1
1                        64512        38325             1:1.68
2             3          64512        40998             1:1.57
3             5          64512        41119             1:1.57
.
.
.
.
56            98         64512        33019             1:1.95
57            100        49344        28747             1:1.72
========== ============ ============ ================= ========
Total:        100        3662016      2002523           1:1.83

FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\file.exe'
FASTCopied to file 'C:\USERS\eliseva\tmp\file.exe ' ( 3662016 Bytes )

Transfer started at : Sun Aug 18 13:21:51 2002
Transfer ended at : Sun Aug 18 13:22:13 2002

FASTCopy termination statistics report:
FASTCopy originally started at : Sun Aug 18 13:21:48 2002
FASTCopy ended at              : Sun Aug 18 13:22:14 2002
```

The following transfer uses the ap compression algorithm:

WINDOWS

```
C:\USERS\eliseva\tmp>fcopy
\\linux\\home\\elish\\tests\\model\\big.file .\\ -report
replace -compression=ap -user=root -password=PASSWORD

########### FASTCopy started at Sun Aug 18 13:03:32 2002########

fcopy “\\\\linux\\\\home\\elish\\tests\\model\\big.file. “.\\”
-report -replace -compression=“ap” -user=“root” -password=PASSWORD -
error_abort

FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD

********** FASTCopy Statistics **********

Local Filename : C:\USERS\eliseva\tmp\big.file
Remote Filename : \\linux\home\elish\tests\model\big.file
Transfer started at : Sun Aug 18 13:03:34 2002

File size : 2817304 bytes

Position    %Completed   Block-size  Compressed-size Ratio
========== ============ ============ ==============  ==========
1              2         64512          14752         1:4.37
2              4         64512          11632         1:5.55
3              6         64512          10997         1:5.87
.
.
.
42             96        64512          3134          1:20.58
43             98        64512          2021          1:31.92
44             100       43288          16190         1:2.67
========== ============ ============ ================ =========
Total:         100       2817304        437943          1:6.43

FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\big.file'
FASTCopied to file 'C:\USERS\eliseva\tmp\big.file' ( 2817304 Bytes )

Transfer started at : Sun Aug 18 13:03:34 2002
Transfer ended at : Sun Aug 18 13:03:51 2002

FASTCopy termination statistics report:

FASTCopy ended at                : Sun Aug 18 13:03:51 2002
```

The following transfer uses the `power` compression algorithm:

WINDOWS

```
C:\USERS\eliseva\tmp>fcopy
\\linux\\home\\elish\\tests\\model\\file.exe .\\ -report=comp.pwr -replace -
compression=power -user=root -password=PASSWORD

########### FASTCopy started at Sun Aug 18 13:22:50 2002#######

fcopy “\\\\linux\\\\home\\elish\\tests\\model\\file.exe. “.\\”
-report=“comp.pwr”-replace -compression=“power” -
user=“root” -password=PASSWORD -error_abort

FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD

FCP-I-LICD FASTCopy demonstration license

********** FASTCopy Statistics **********


Local Filename  : C:\USERS\eliseva\tmp\file.exe
Remote Filename : \\linux\home\elish\tests\model\file.exe
Transfer started at : Sun Aug 18 13:22:52 2002

File size : 3662016 bytes
Position    %Completed   Block-size  Compressed-size  Ratio
========== ============ ============ =============== ==========
1               1          64512            27333      1:2.36
2               3          64512            29289      1:2.20
3               5          64512            29139      1:2.21
.
.
.
56              98         64512            23091      1:2.79
57              100        49344            20158      1:2.45
========== ============ ============ =============== ===========
Total:          100        3662016          1436415     1:2.55

FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\file.exe'
FASTCopied to file 'C:\USERS\eliseva\tmp\file.exe' ( 3662016 Bytes )

Transfer started at : Sun Aug 18 13:22:52 2002
Transfer ended at : Sun Aug 18 13:25:16 2002

FASTCopy termination statistics report:
FASTCopy originally started at : Sun Aug 18 13:22:50 2002
FASTCopy ended at              : Sun Aug 18 13:25:16 2002
```

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

COMPRESSED_SUFFIX

Syntax: `compressed_suffix=`*`“x[, yy, z, ...]”`*

The `compressed_suffix` qualifier allows you to specify which file types FASTCopy should not compress. These files should have names in the format **filename.*tag***, where ***tag*** is an extension similar to those used by most common compression utilities. For example, files compressed with the **GZIP** program are appended with the extension: ***gz***.

Attempting to compress files a second time rarely reduces file size. Files specified by the `compressed_suffix` filter are excluded from the compression phase of the FASTCopy transfer, thereby reducing overall transfer time.

Attributes

LIST_VALUE
OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

INTERVAL

Syntax: `interval=`*n*

The `interval` qualifier specifies the interval in seconds that FASTCopy should wait before sending the next buffer. The interval qualifier is sometimes used to reduce the network load for interactive users. The default value of interval is `0` (no interval between packets).

Note: The `interval` filter slows down file transfer and should be used only if you wish to deliberately slow down FASTCopy in order to reduce the network load.

See also: `ACK_INTERVAL` (115).

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

MAX_SMALL_FILE

Syntax: `max_small_file=`*`bytes`*

The `max_small_file` qualifier specifies the maximum size in bytes of files that FASTCopy will consider "small ". The default value of this qualifier is 1 (all files containing data are considered "large") and can be set to a maximum of 45,000 bytes. Files designated as "small" by this qualifier are transferred by FASTCopy using an abbreviated protocol which is more efficient than the standard FASTCopy protocol. When files designated as "small" are transferred, the `model` qualifier is ignored. If you specify a model file when transferring a source file smaller than the `max_small_file` value, the file will always be transferred without using the model file. This is true even if the `min_model_size` is set below the `max_small_file` value.

When working over high-speed communications lines, it is recommended that the `max_small_file` value be set to 45,000 (the maximum) for more efficient transfer of smaller files. When the transfer rates are low and model is used to transfer "small" files, the value of `max_small_file` should be decreased accordingly.

See also: `MIN_MODEL_SIZE` (129), `MODEL` (130).

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

MAX_WINDOW_SIZE

Note: This qualifier is supported from version 2.6.5 upwards.

Syntax: `max_window_size=`*`numeric_value`*
Default: `max_window_size=3`

The `max_window_size` qualifier sets an upper limit for the number of packets that the source node can send without receiving an acknowledgement from the target node. By default, the source node waits for an "acknowledgement" from the target node after every three packets. In large files, this may have a negative impact on the overall transfer speed. In most cases, increasing the number of packets required before each acknowledgement will increase the overall transfer speed.

See also: `ACK_INTERVAL` (115)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

MIN_MODEL_SIZE

Syntax: `min_model_size=`*`bytes`*

The `min_model_size` qualifier specifies the minimum file size for which the model option is activated. This is useful when you are transferring multiple files, and you only wish the larger files to be transferred using a model file. The default value is 0, meaning that the model option will be used for files of any size. The `min_model_size` qualifier must be used with the `MODEL` qualifier.

The `min_model_size` qualifier is overridden when the `MAX_SMALL_FILE` qualifier is set to a higher value.

See also: `MAX_SMALL_FILE` (127), `MODEL` (130).

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

MODEL

Syntax: `model`*`[=model_file_name]`*
Default: `model`*`[=target_file_name]`*

The `model` qualifier specifies a file on the target node with contents close to the source file that you are transferring. FASTCopy dynamically compares the source file to the model file during the transfer. When FASTCopy finds identical data in these files, instead of transferring the data from the source over the network, it takes the data from the model file (locally on the target system) to the target file.

The result of this process is that only the differences between the source file on the source system and the model file on the target system are transferred over the network.

If the `model` qualifier is specified without a *`model_file_name`*, the target file itself is taken as the model file for the transfer. In this case, the `temp_dir` qualifier must be specified.

The `model` qualifier does not apply to files smaller than the value of the `max_small_file` qualifier, if you specify that qualifier.

See also: `MIN_MODEL_SIZE` (129)

Example 1:

In the initial transfer FASTCopy copies the file **a.a** from the current working directory to the file **/usr/develop/b.b** on the remote node **hipi**. The file size is **212992** bytes.

UNIX

```
sun:/usr/develop> fcopy a.a hipi:/usr/develop/b.b -report
-compression

fcopy a.a hipi:/usr/develop/b.b -report
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
********************** FASTCopy Report ***********************
Local Filename  : /usr/develop/a.a
Remote Filename : hipi:/usr/develop/b.b
FASTCopy started at : Sun Jan 7 15:15:00 2002

File size : 212992 bytes

Position   %Completed  Block-size  Compressed-size Ratio
========== =========== =========== =============== ===========
1              30       64512        37139          1:1.74
2              60       64512        36820          1:1.75
3              90       64512        64512          1:1.00
4              100      19456        5561           1:3.50
========== =========== =========== =============== ===========
Total:         100      212992       144032         1:1.48

FCOPY-S-FILE_COPIED - File '/usr/develop/a.a' FASTCopied to file
'hipi:/usr/develop/b.b' ( 212992 Bytes )

FASTCopy started at : Sun Jan 7 15:15:00 2002
FASTCopy ended at : Sun Jan 7 15:15:07 2002
```

Example 2:

In the following transfer, the user makes a slight change to the source file by including additional data. Since the source file is basically the same as the target file, the user instructs FASTCopy to transfer the updated source file **a.a** to **/usr/develop/c.c** on the target node **hipi**, using the previously transferred file - **/usr/develop/b.b** - as the model file. During the transfer FASTCopy will attempt to copy as much data as possible from that file instead of transferring it over the network.

UNIX

```
sun:/usr/develop> echo “one last record” >> a.a
sun:/usr/develop> fcopy a.a hipi:/usr/develop/c.c -report
-model=/usr/develop/b.b
```

The results of this transfer are:

UNIX

```
fcopy a.a hipi:/usr/develop/c.c -report -compression
-model=/usr/develop/b.b

############ FASTCopy started at Wed Feb 21 12:12:38 2002######

FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD

********** FASTCopy Statistics **********

FCOPY-I-OPEN_MODEL_S - Model file '/usr/develop/b.b' opened
Local Filename  : /usr/develop/a.a
Remote Filename : hipi:/usr/develop/c.c
FASTCopy started at : Sun Jan 7 15:16:23 2002
```

```
File size : 213008 bytes

Position   %Completed   Block-size Compressed-size  Ratio
========== =========== =========== =============== ===========
1              30         64512        12          1:5376.00 *M*
2              60         64512        12          1:5376.00 *M*
3              90         64512        12          1:5376.00 *M*
4              100        19472        5573        1:3.49
========== =========== =========== =============== ===========
Total:         100        213008       5609          1:37.98

FCOPY-S-FILE_COPIED - File '/usr/develop/a.a' FASTCopied to file
'hipi:/usr/develop/c.c' ( 213008 Bytes )

FASTCopy started at : Sun Jan 7 15:16:23 2002
FASTCopy ended at : Sun Jan 7 15:16:24 2002
```

FASTCopy copies the first three blocks of data from the model file (in this case the target file) on the node **hipi**. Only the last block containing the new data is transferred over the network. Thus, instead of sending the full 213008 bytes over the network, only 5609 bytes are transferred.

Example 3:

Sometimes, especially when transferring multiple files, you may wish to use the existing output file as a model file rather than specifying a specific model file. If this is the case, specify the model qualifier without a value and provide a temporary directory for the output file (using the `temp_dir` qualifier). The file will be copied to the temporary directory and given a temporary name, using the existing target as the model file. At the end of transfer, the temporary file replaces the output file and is renamed accordingly.

The next example transfers again the source file **a.a** without a change to update the target file **/usr/develop/c.c** on node **hipi**. The files are already identical before the transfer. The example will use the actual target file as the model file.

UNIX

```
sun:/usr/develop> fcopy a.a hipi:/usr/develop/c.c -model
-temp_dir=/tmp -report – compression

fcopy a.a hipi:/usr/develop/c.c -model -temp_dir=/tmp -report
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
***********************FASTCopy Report ***********************
FCOPY-I-OPEN_MODEL_S - Model file '/usr/develop/c.c' opened
Local Filename : /usr/develop/a.a
Remote Filename : hipi:/usr/develop/c.c
Temporary output file : hipi:/tmp/AAAa11777
```

```
FASTCopy started at : Sun Jan 7 15:17:10 2002
File size : 213008 bytes
```

The file is copied to a temporary output file **/tmp/AAAa11777**, using **/usr/develop/c.c** as the model. Since the source file and the model file are identical, the transfer report will appear as follows:

UNIX

```
Position  %Completed Block-size  Compressed-size  Ratio
========= ========== =========== ================ =========
1             30       64512         12           1:5376.00 *M*
2             60       64512         12           1:5376.00 *M*
3             90       64512         12           1:5376.00 *M*
4             100      19472         12           1:1622.67 *M*
========= ========== =========== ================ ==========
Total:        100      213008        48           1:4437.67

FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/tests/src/a.a '
FASTCopied to file 'hipi:/tmp/AAAa11777' ( 213008 Bytes )

FASTCopy started at : Sun Jan 7 15:17:10 2002
FASTCopy ended at : Sun Jan 7 15:17:10 2002

FCOPY-I-RENAME - Temporary output file '/tmp/AAAa11777' renamed to
'/usr/develop/c.c'
```

You can see that the entire file was copied from the model file. Instead of **213008** bytes, only **48** bytes were transferred over the network. At the end of transfer, the temporary file **/tmp/AAAa11777** was renamed to replace the target file **/usr/develop/c.c**.

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

MODEL_TABLE_SIZE

Note: This qualifier is supported from version 2.6.5 upwards.

Syntax: `model_table_size=numeric_value`
Default: `model_table_size=7500`

The `model_table_size` qualifier - supported from FASTCopy V2.6.5 upwards - allows you to configure the number of packets that will be compared in each model pass. The default and maximum permitted size of the model CRC table is 7500 entries, allowing for rapid comparison of the source and model files. However, a CRC table with 7500 entries consumes greater CPU than a CRC table with fewer entries. Depending on the CPU of the machines involved in the transfer, you may wish to specify less than 7500 entries.

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

Filter Qualifiers

This section details qualifiers that filter the files included in the file specification. For example, excluding specific files or file types, or selecting files according to their file attributes.

Filter qualifiers include:

BEFORE

Syntax: `before[=`*`time`*`]`

The `before` qualifier selects only those files that are dated before the specified *time*.

The `before` qualifier can only be used with the `changed_status`, `created` or `modified` qualifiers. In UNIX, Windows NT and Windows 9x/2000/XP systems, `modified` is the default, while in OpenVMS systems, `created` is the default.

See Appendix A: Setting Time and Time-Frame Expressions (285). for a description of valid time expression syntax.

See also: `CHANGED_STATUS` (138), `CREATED` (139), `MODIFIED` (143), and `SINCE` (147).

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

CHANGED_STATUS

Syntax: `changed_status`

The `changed_status` qualifier selects files according to their last status change date.

This qualifier can only be used with the `before` or `since` qualifiers, as an alternative to the `created` or `modified` qualifiers (the UNIX default).

See also: `BEFORE` (137), `CREATED` (139) `MODIFIED` (143), and `SINCE` (147).

Attributes

POSITIONAL

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

CREATED

Syntax: `created`

The `created` qualifier selects files according to their creation date.

This qualifier can only be used with the `before` or `since` qualifiers, as an alternative to the `modified` qualifier.

On OpenVMS systems, `created` is the default, while on Windows systems, `modified` is the default.

See also: `BEFORE` (137), `CHANGED_STATUS` (138), `MODIFIED` (143), and `SINCE` (147).

Attributes

POSITIONAL

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	Yes

EXCLUDE

Syntax: `exclude=`*`filespec`*

The `exclude` qualifier prevents files that match the specified file specification from being transferred to the target node. The `exclude` qualifier's value - *`filespec`* - is a file specification in the same format as the source and destination file specifications. The `exclude` qualifier is useful in cases when it is more convenient to tell FASTCopy what files you wish to exclude from the transfer, then to specify which files you wish to transfer.

Example 1:

Alice has different types of files in the directory **c:\develop\bin** - source files, resource files, data files, and executables. She wants to move all of these files except for the executables and the dynamic link library files into a common directory used by her colleagues, on a common UNIX server called **bob**:

WINDOWS

```
C:\DEVELOP> fcopy bin\*.* \\bob\developers\project\source\*.*
-exclude=“*.dll, *.exe”
```

Example 2:

Webmaster@poseur.com decides to copy the company intranet site from the UNIX server **oz** to directory **c:/web** on his own Windows NT computer at home, so that he can tinker with it at his leisure. The intranet site has a mixture of **HTML**, **JPEG**, **GIF**, **WAV**, **MIDI** and **Java** files, all of which he can view locally in his browser; but it also has some heavy server **CGI** programs that were compiled on UNIX. To prevent these programs from taking up space on his disk, he uses this command:

WINDOWS

```
c:> fcopy \\oz.poseur.com\www\*.* c:\\web\*.* -recursion
-user=webmaster -password=fTYgS%$nh -exclude=“*.cgi”
```

Attributes

POSITIONAL
OVERRIDE
LIST VALUE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

GID

Syntax: `gid`*`[=GID]`*

The `gid` qualifier selects files only if their owners' Group Identification (GID) matches the specified *GID*.

See also: `UID` (150)

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

MODIFIED

Syntax: `modified`

The `modified` qualifier selects files according to the dates that they were last modified.

This qualifier can only be used with the `before` or `since` qualifiers, as an alternative to the `created` (on OpenVMS, Windows NT and Windows 9x/2000/XP) or `changed_status` (on UNIX) qualifiers.

On UNIX, Windows NT and Windows 9x/2000/XP systems, `modified` is the default, while in OpenVMS systems, `created` is the default.

See also: `BEFORE` (137), `CHANGED_STATUS` (138), `CREATED` (139) and `SINCE` (147).

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

OWNER_GROUP

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax: `owner_group=`*`group_name`*

The `owner_group` qualifier instructs FASTCopy to transfer only those files that match the *group name* (as opposed to the `gid` qualifier, which instructs FASTCopy to transfer only those files that match the *`goup ID`*).

Example:

```
fcopy abc\* 192.147.160.168:/dev/d/qajoel/ -user=administrator -pass=XX -repo
-repl -owner_group=qa
```

See also: `GID` (142)

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

OWNER_USER

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax: `owner_user=`*`user_name`*

The `owner_user` qualifier instructs FASTCopy to transfer only those files that match the *username* (as opposed to the `uid` qualifier, which instructs FASTCopy to transfer only those files that match the *`user ID`*).

Example:

```
fcopy abc\* 192.147.160.168:/dev/d/qajoel/ -user=administrator -pass=XX -repo
-repl -owner_user=root
```

See also: `UID` (150)

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

PRIORITY

Syntax: `priority=numeric_value`

The `priority` qualifier allows you to submit FASTCopy jobs according to their order of importance. For example, if you have 8 jobs scheduled to run at the same time, the Job with `priority=1` will be submitted first and the Job with `priority=8` will be submitted last.

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

SINCE

Syntax: `since`*`[=time]`*

The `since` qualifier selects only those files that are dated after the specified time.

The `since` qualifier can only be used with the `changed_status`, `created` or `modified` qualifiers. On UNIX, Windows NT and Windows 9x/2000/XP systems, `modified` is the default, while on OpenVMS systems, `created` is the default.

See Appendix A: Setting Time and Time-Frame Expressions (285) for a description of valid time expression syntax.

See also: `BEFORE` (137), `CHANGED_STATUS` (138), `CREATED` (139) and `MODIFIED` (143).

Example

The following is a directory listing of the **linux/tmp** directory on the UNIX node:

UNIX

```
linux/tmp> ls -l
drwxrwxrwx   4 root   root      1024 Aug 18 13:02 ./
drwxr-xr-x   8 elish  users     1024 Aug 7 11:20 ../
-rw-r-r-   1 root   root     23641 Jul 16 14:15 a.a
-rw-r-r-   1 root   root     23641 Aug 1 14:53 a.today
-rw-r-r-   1 root   root     23641 Jul 30 12:43 b.b
-rw-r-r-   1 root   root     23641 Aug 1 14:53 b.today
-rw-r-r-   1 root   root   2817304 Aug 18 12:44 big.file
-rwxr-xr-x   1 root   root   3662016 Aug 18 13:02 file.exe*
```

On a networked Windows NT node, the user enters the following command to transfer all files that were modified since the 18th of August, 2002 from the directory **linux/tmp** on the UNIX node to the local directory **\test** on the Windows NT node. As the modified action is the default on the source node (UNIX), the `modified` qualifier is not required and is included only to clarify the example.

WINDOWS

```
c:> fcopy \\linux\\tmp\*.* test\ -modified -since=180802 -user=root -
password=PASSWORD -replace
FCOPY-S-FILE_COPIED - File  '\\linux\home\elish\tests\model\big.file '
```

```
FASTCopied to file 'C:\USERS\eliseva\tmp\big.file ' ( 2817304 Bytes )
FCOPY-S-FILE_COPIED - File '\\linux\home\elish\tests\model\file.exe '
FASTCopied to file 'C:\USERS\eliseva\tmp\file.exe ' ( 3662016 Bytes )
FCOPY-S-FINISHED, FASTCopy operation successfully finished
```

The following is the directory listing of the directory **\test** on the target node:

WINDOWS

```
Volume in drive C is unlabeled Serial number is 946F:099A
Directory of C:\USERS\eliseva\tmp/*.*
8/18/02   15:06           <DIR>   .
8/18/02   15:06           <DIR>   ..
8/18/02   15:06         2,817,304 big.file
8/18/02   15:06         3,662,016 file.exe
     6,479,320 bytes in 2 files 6,576,128 bytes allocated
    2,210,816 bytes free
```

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

UIC

Syntax: `uic[=`*`UIC`*`]`

The `uic` qualifier selects files only if their owners' User Identification Code (UIC) matches the specified owner *UIC*.

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	No
OpenVMS	Yes

UID

Syntax: `uid[=`*`UID`*`]`

The `uid` qualifier selects files only if their owners' User Identification (UID) matches the specified owner *UID*.

See also: `GID` (142)

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

Recovery Qualifiers

This section details qualifiers that control the recovery of failed FASTCopy operations.

Recovery qualifiers include:

- `BASE_INTERVAL` (152)
- `FACTOR` (153)
- `FORCE` (154)
- `MAX_INTERVAL` (158)
- `RECOVER` (159)
- `RETRIES` (164)
- `SKIP` (165)

BASE_INTERVAL

Syntax: `base_interval=`*`minutes`*

The `base_interval` qualifier specifies the time in minutes between recovery attempts of an aborted FASTCopy operation running in batch mode.

See also: `FACTOR` (153), `MAX_INTERVAL` (158) and `BATCH` (204).

Attributes

OVERRIDE
BATCH
WINDOW

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

FACTOR

Syntax: `factor=`*`percentage`*

The `factor` qualifier determines the factor by which the base interval is extended after each recovery attempt made on a FASTCopy operation running in batch model. The factor is expressed as a percentage of the base interval.

For example, if `factor=20` and `base_interval=5`, then:

- The time between the first abort and the first recovery will be 5 minutes;
- The time between the second abort and the second recovery will be 5 + 20%*5 = 6 minutes;
- The time between the third abort and the third recovery will be 6 + 20%*6 = 7 minutes and 12 seconds;

And so on.

If `factor=0` the base interval will not change (this is the default when the `factor` qualifier is not specified).

If `factor=100` then the interval is doubled for each subsequent recovery attempt.

See also: `BASE_INTERVAL` (152) and `MAX_INTERVAL` (158)

Attributes

OVERRIDE
BATCH
WINDOW

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

FORCE

Syntax: `force`

Note: As of FASTCopy version 2.3.1, the force qualifier has been replaced with the new force value of the recover qualifier. The qualifier is still supported so that old scripts will continue to work, but its use is now depreciated.

The `force` qualifier forces FASTCopy to attempt recovery, even after a non recoverable error occurs. For example, if a mismatch is discovered between the source and destination files during a recovery attempt, and your command line includes the `force` qualifier, FASTCopy will attempt to resend the aborted file. If the `force` qualifier (the default situation) is not specified, FASTCopy aborts the recovery process with an error message.

See also: `RECOVER` (159), `SKIP` (165).

Example

The following example illustrates the outcome of a mismatch between the source file and the partial destination file.

The original command is aborted in the middle of the transfer.

UNIX

```
>fcopy a.a vinto:/tmp -report -verify=“sync,crc,compare” -replace
-error_abort –compression

############ FASTCopy started at Wed Feb 21 12:12:38 2002 ###########

FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD

********** FASTCopy Statistics **********
Local Filename : /usr/develop/yossi/fastcopy/src/a.a
Remote Filename : vinto:/tmp/a.a
FASTCopy started at : Wed Feb 7 12:16:11 2002
File size : 258164 bytes
Position    %Completed   Block-size   Compressed-size   Ratio
========== ============ ============ ================= ===========
1               24         64512       28856           1:2.24
2               49         64512       28778           1:2.24
3               74         64512       26627           1:2.42
4               99         64512       29450           1:2.19
5               100        116         23              1:5.04
^C
```

```
>
```

On the target node **vinto**, the partial target file is corrupted by the following command:

UNIX

```
vinto > echo “blabla” > /tmp/a.a
```

When normal recovery is attempted, FASTCopy detects a mismatch:

UNIX

```
/usr/develop/yossi/fastcopy/src_fcopy –recover

############ FASTCopy started at Wed Feb 21 12:12:38 2002 ##########

FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
FCOPY-I-RECOVER - Starting recovery operation
FCOPY-I-RECOVER_TIME - Wed Feb 7 12:16:56 2002
FCOPY-I-CMD_LINE - Original command was - 'fcopy a.a vinto:/tmp
-report -verify=“sync,crc,compare” -replace -error_abort
-compression'

********** FASTCopy Statistics **********
fcopy -recover=“recover” -error_abort
Local Filename : /usr/develop/yossi/fastcopy/src/a.a
Remote Filename : vinto:/tmp/a.a
FASTCopy started at : Wed Feb 7 12:16:11 2002
File size : 258164 bytes
Position    %Completed   Block-size   Compressed-size   Ratio
========== ============ ============ ================= ===========
FCOPY-I-ANALYZE_START - Starting partial target file analyze ...
FCOPY-E-VERIFY, Input file /usr/develop/yossi/fastcopy/src/a.a does
not match output file vinto:/tmp/a.a
```

When the `force` qualifier is used together with the `recover` qualifier, FASTCopy attempts to recover from this mismatch. It tries to find the first match between the partial target file and the source file and, on finding a match, recovers from that point.

UNIX

```
> fcopy -recover -force
fcopy -recover=“recover” -force -error_abort
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
FCOPY-I-RECOVER - Starting recovery operation
FCOPY-I-RECOVER_TIME - Wed Feb 7 12:17:12 2002
FCOPY-I-CMD_LINE - Original command was - 'fcopy a.a vinto:/tmp
-report -verify=“sync,crc,compare” -replace -error_abort
-compression'

********** FASTCopy Statistics **********
Local Filename : /usr/develop/yossi/fastcopy/src/a.a
Remote Filename : vinto:/tmp/a.a
FASTCopy started at : Wed Feb 7 12:16:11 2002
File size : 258164 bytes
Position    %Completed   Block-size   Compressed-size   Ratio
========== ============ ============ ================ ===========
FCOPY-I-ANALYZE_START - Starting partial target file analyze ...
FCOPY-E-VERIFY, Input file /usr/develop/yossi/fastcopy/src/a.a does
not match output file vinto:/tmp/a.a
```

After a mismatch is found, FASTCopy tests if a smaller part of the target file matches the source file. Since in this example the entire file is corrupted, FASTCopy resets to the beginning of the file and recovers from there.

UNIX

```
FCOPY-I-RESET - reanalyzing first 258048 bytes of file
FCOPY-E-VERIFY, Input file /usr/develop/yossi/fastcopy/src/a.a does
not match output file vinto:/tmp/a.a
FCOPY-I-RESET - reanalyzing first 193536 bytes of file
FCOPY-E-VERIFY, Input file /usr/develop/yossi/fastcopy/src/a.a does
not match output file vinto:/tmp/a.a
FCOPY-I-RESET - reanalyzing first 129024 bytes of file
FCOPY-E-VERIFY, Input file /usr/develop/yossi/fastcopy/src/a.a does
not match output file vinto:/tmp/a.a
FCOPY-I-RESET - reanalyzing first 64512 bytes of file
FCOPY-E-VERIFY, Input file /usr/develop/yossi/fastcopy/src/a.a does
not match output file vinto:/tmp/a.a
```

UNIX

```
FCOPY-I-RESET - reanalyzing first 0 bytes of file
1            24           64512           28856          1:2.24
2            49           64512           28778          1:2.24
3            74           64512           26627          1:2.42
4            99           64512           29450          1:2.19
5            100          116             23             1:5.04
========== ============ ============ ================= ===========
Total: 100 258164 227468 1:1.13

FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/fastcopy/src/a.a '
FASTCopied to file 'vinto:/tmp/a.a' ( 258164 Bytes )

FASTCopy started at : Wed Feb 7 12:16:11 2002
FASTCopy ended at : Wed Feb 7 12:17:17 2002

FCOPY-I-VERIFY_START - Starting verification pass ...
FCOPY-I-VERIFY_OK - Files compared successfully

FASTCopy termination statistics report:
FASTCopy originally started at : Wed Feb 7 12:16:11 2002
FASTCopy ended at              : Wed Feb 7 12:17:17 2002
>
```

Attributes

RECOVER ONLY

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

MAX_INTERVAL

Syntax: `max_interval=`*`minutes`*

The `max_interval` qualifier determines the maximum size in minutes of the interval between recovery attempts of an aborted FASTCopy operation running in batch mode. This interval starts at the value specified by the `base_interval` qualifier and will increase in size from one attempt to the next if the `factor` qualifier is specified. Specifying a `max_interval` value places an upper limit on this growth.

See also: `BASE_INTERVAL` (152), `FACTOR` (153)

Attributes

BATCH
WINDOW

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	Yes

RECOVER

Syntax: `recover[=recover_type]`
Default: `recover=recover`

The `recover` qualifier recovers a previously interrupted FASTCopy operation.

The *`recover_type`* can be `recover`, `restart`, `skip`, or `force`.

recover	Performs recovery from the point of failure.
restart	FASTCopy restarts the operation.
skip	During a multiple file transfer operation, `skip` instructs FASTCopy to resume the transfer, beginning with the source file that follows the aborted one. Use this value in situations where a file cannot be accessed for some reason (such as a read error on the file) and you decide to continue the operation from the next file because it is clear to you that recovery will not help.
force	During a multiple file transfer operation, `force` compels FASTCopy to attempt recovery, even after a non recoverable error. If a mismatch is discovered between the source and destination files during the recovery attempt, FASTCopy will restart transmission of the disrupted file.

The recoverability of a FASTCopy operation is made possible by the operation's details and status being stored in a context file. The default FASTCopy context file is **fcopy_context.dat.*nnnnn*** in the user's home directory where ***nnnnn*** is the highest context file version number in the directory. On OpenVMS platforms, version numbering is supported internally, so the OpenVMS version number replaces the ***.nnnnn*** suffix.

You can change the default context filename by specifying a different name with the environment variable *FCOPY_CONTEXT_FILE* (on UNIX, Windows 9x and Windows NT), or with the OpenVMS logical name: FASTCOPY$CONTEXT_FILE. You can override the default context file by using the `context_file` qualifier with the `recover` qualifier and specifying a different filename.

See also: `CONTEXT_FILE` (22), `DUMP` (28), `FORCE` (154) and `SKIP` (165).

Example

In the following example, the transfer operation of a single file is aborted and then restarted by issuing another FASTCopy command with the `recover` qualifier.

UNIX

```
> fcopy a.a vinto:/tmp -report -verify -replace
fcopy a.a vinto:/tmp -report -verify=“sync,crc,compare” -replace
-error_abort -compression
########## FASTCopy started at Wed Feb 21 12:14:38 2002 ##########

FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD

***************** FASTCopy Statistics ******************
Local Filename : /usr/develop/yossi/fastcopy/src/a.a
Remote Filename : vinto:/tmp/a.a

File size : 258164 bytes

Position   %Completed   Block-size Compressed-size Ratio
========= =========== =========== ================ =========
1            24          64512       28856          1:2.24
2            49          64512       28778          1:2.24
3            74          64512       26627          1:2.42
^C
>
```

The transfer was aborted (manually) after 74% of the file was transferred. To recover from this operation and continue the transfer from the point of failure, the user should issue another FASTCopy command with the `recover` qualifier.

The recovery process begins with an analysis phase during which the source file and the partial destination file are compared to make sure that they are identical.

UNIX

```
> fcopy -recover
fcopy -recover=“recover” -error_abort
##########FASTCopy started at : Wed Feb 7 12:14:38 2002##########
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
FCOPY-I-RECOVER - Starting recovery operation
FCOPY-I-RECOVER_TIME - Wed Feb 7 12:15:07 2002
FCOPY-I-CMD_LINE - Original command was - 'fcopy a.a vinto:/tmp
-report -verify=“sync,crc,compare” -replace -error_abort
-compression'
******************* FASTCopy Statistics ******************
Local Filename : /usr/develop/yossi/fastcopy/src/a.a
Remote Filename : vinto:/tmp/a.a
File size : 258164 bytes
Position    %Completed   Block-size  Compressed-size   Ratio
========== ============ ============ ================= ===========

FCOPY-I-ANALYZE_START - Starting partial target file analyze ...
FCOPY-I-ANAL_OK - First 193536 bytes of target file match source file
```

Only after successful verification is the transfer resumed from the point of failure. The verification that was requested at the original command will be issued after the completion of the transfer.

UNIX

```
4              99        64512           29753         1:2.17
5              100       116             23            1:5.04
========== ============ ============ ================= ===========
Total:         100       258164          114037        1:2.26

FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/fastcopy/src/a.a '
FASTCopied to file 'vinto:/tmp/a.a' ( 258164 Bytes )

FASTCopy started at : Wed Feb 7 12:14:38 2002
FASTCopy ended at : Wed Feb 7 12:15:08 2002

FCOPY-I-VERIFY_START - Starting verification pass ...
FCOPY-I-VERIFY_OK - Files compared successfully
```

The following example shows recovery from aborted operation with the restart option and overriding the original `compression` qualifier.

UNIX

```
>fcopy a.a vinto:/tmp -report -verify=“sync,crc,compare” -replace
-error_abort -compression
##########FASTCopy started at : Wed Feb 7 12:15:28 2002##########
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
***************** FASTCopy Statistics *****************
Local Filename : /usr/develop/yossi/fastcopy/src/a.a
Remote Filename : vinto:/tmp/a.a
File size : 258164 bytes
Position    %Completed   Block-size   Compressed-size  Ratio
========== ============ ============ ================= ===========
1              24        64512         28856           1:2.24
2              49        64512         28778           1:2.24
3              74        64512         26627           1:2.42
^C
> fcopy -recover=restart -compression=none
fcopy -recover=“restart” -compression=“none” -error_abort
##########FASTCopy started at : Wed Feb 7 12:15:48 2002##########
FCOPY-I-CPYRGHT - FASTCopy V2.6 Copyright (c) 1990-2002 by SoftLink LTD
FCOPY-I-RECOVER - Starting recovery operation
FCOPY-I-RECOVER_TIME - Wed Feb 7 12:15:48 2002
FCOPY-I-CMD_LINE - Original command was - 'fcopy a.a vinto:/tmp
-report -verify=“sync,crc,compare” -replace -error_abort -compression'
```

UNIX

```
***************** FASTCopy Statistics *****************
Local Filename  : /usr/develop/yossi/fastcopy/src/a.a
Remote Filename : vinto:/tmp/a.a
FASTCopy started at : Wed Feb 7 12:15:48 2002
File size : 258164 bytes
Position    %Completed  Block-size   Compressed-size   Ratio
========== ============ ============ ================= ===========
1           24          64512        64512             1:1.00
2           49          64512        64512             1:1.00
3           74          64512        64512             1:1.00
4           99          64512        64512             1:1.00
5           100         116          116               1:1.00
========== ============ ============ ================= ===========
Total:      100         258164       258164            1:1.00
FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/fastcopy/src/a.a '
FASTCopied to file 'vinto:/tmp/a.a' ( 258164 Bytes )
FASTCopy started at : Wed Feb 7 12:15:48 2002 FASTCopy ended at :
Wed Feb 7 12:15:49 2002
FCOPY-I-VERIFY_START - Starting verification pass ...
FCOPY-I-VERIFY_OK - Files compared successfully
```

Attributes

NO_PARAMETERS
RECOVER_ONLY

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

RETRIES

Syntax: `retries=`*`max_retries`*

The `retries` qualifier specifies the maximum number of retry attempts for a FASTCopy operation executed in batch mode. If this number is reached, the operation is terminated with an error.

See also: `BATCH` (204).

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

SKIP

Syntax: `skip`

Note: As of FASTCopy version 2.3.1, the skip qualifier has been replaced with the new skip value of the recover qualifier. The qualifier is still supported so that old scripts will continue to work, but its use is now depreciated.

During a multiple file transfer operation, the `skip` qualifier instructs FASTCopy to resume the transfer, beginning with the source file that follows the aborted one. The `skip` qualifier skips the current file only if the following conditions are met:

1. It is specified with the `recover=recover` (default) qualifier.
2. The transfer is aborted in the middle of a file.

If the transfer was NOT aborted in the middle of a file, for example between files or at the beginning (no connection), the `skip` qualifier will be ineffective.

This qualifier was added to handle situations where a file cannot be accessed for some reason (such as a read error on the file) and the user decides to continue the operation from the next file because it is clear that recovery will not help.

See also: `FORCE` (154), `RECOVER` (159).

Attributes

RECOVER_ONLY

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

Encryption Qualifiers

Note: In order to perform encrypted transfers, you must have a license that supports encryption installed on both the local and the remote nodes.

This section details qualifiers that are used to implement security measures in transfer operations, such as data encryption, partner authentication, and FASTCopy Proxy Security. Encryption qualifiers are only effective if matching settings are defined in the SoftLink **Security and Administration Files**. For further information about the **Security and Administration Files**, refer to the *FASTCopy Administrator's Guide*.

For information on SSL 3.0, see General SSL Qualifiers (177).

Security qualifiers include:

- `CODE` (167)
- `LINE_CIPHER` (168)
- `LINE_ENCRYPT` (170)
- `LINE_PHRASE` (172)
- `MAC` (174)
- `SL_SECURITY` (176)

CODE

Syntax: `code=`*`code`*

The `code` qualifier is used to specify an agreed-upon password included with the command. The **Security and Administration Files** used to set permissions for FASTCopy file transfer operations can require that the user gives a special password to carry out a certain operation, such as transferring a specific file or the contents of a specific directory. The `code` qualifier can be used for ad-hoc security, especially at the level of single files or directories.

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LINE_CIPHER

Syntax: `line_cipher[=`*`encryption_algorithm`*`]`

The `line_cipher` qualifier determines what encryption method should be used to encrypt files transferred during an operation. Only the data (i.e. the files) are encrypted by this encryption method; in particular, messages sent back and forth between the FASTCopy application and the FASTCopy daemon on the remote node are not encrypted.

The `line_cipher` qualifier does not make FASTCopy encrypt the transferred data; the `LINE_ENCRYPT` qualifier is used to force that action. The `line_cipher` qualifier only tells FASTCopy what method to use when encrypting a file. A third qualifier - line_phrase - is required to specify which key FASTCopy should use for the encryption.

Currently, the *`encryption_algorithm`* value can be one of the following:

`des, des-cbc, des3, des-ede3, des-ede3-cbc, des-ede, des-ede-cbc, des-ecb, des-cfb, des-ofb, des-ede-ofb, des-ede-cfb, des-ede3-ofb, des-ede3-cfb, desx, desx-cbc, bf-cbc, bf-cfb, bf-ecb, bf-ofb, cast-cbc, cast5-cbc, cast5-cfb, cast5-ecb, cast5-ofb, rc2-40-cbc, rc2-64-cbc, rc2-cbc, rc2-cfb, rc2-ecb, rc2-ofb and rc4-40.`

For each method, the full name should be used rather than an abbreviation. Some of these values are synonyms that specify the same encryption method (for example, specifying `des` is the same as specifying `des-cbc`). See Appendix B: Encryption Methods (290), for an explanation of the various encryption methods available.

If the `line_cipher` qualifieris not specified or is used without a value, FASTCopy assumes that when `line_encrypt` is specified, the `des-cbc` encryption method should be used. If `line_encrypt` *is* specified, FASTCopy will carry it out only if the cipher and phrase specified in the operation's command line match the cipher and phrase specified in the remote node's **Security and Administration Files**. However, it is sufficient that one side specifies `line_encrypt` for data encryption to take place.

See `LINE_ENCRYPT` (170) and `LINE_PHRASE` (172), as well as the FASTCopy *Administrator's Guide* for details on how to set the encryption parameters in the **Security and Administration Files** on the receiving node.

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LINE_ENCRYPT

Syntax: `line_encrypt`

The `line_encrypt` qualifier encrypts one or more of the files transferred in the operation, using the key specified by the `line_phrase` qualifier and the encryption method specified with the `LINE_CIPHER` qualifier. (If no method is specified, FASTCopy assumes the encryption method to be `des-cbc`). Note that FASTCopy will only perform the transfer if the cipher and phrase specified in the command match those specified in the receiving node's ***Security and Administration Files***. Otherwise, an error occurs. Only the data (i.e. the files) are encrypted during the operation; in particular, messages sent back and forth between the FASTCopy application and the FASTCopy daemon on the remote node are not encrypted.

When the `line_encrypt` qualifier is used, it must be accompanied with a key, specified by the `line_phrase` qualifier. If the default encryption method is not used, the desired encryption method should also be specified, using the `line_cipher` qualifier.

While the cipher and phrase must be specified both in the operation's command line and the remote node's ***Security and Administration Files***, specifying `line_encrypt` in either one of these places will cause files to be encrypted.

See Appendix B: Encryption Methods (290), for an explanation of the various encryption methods available.

See `LINE_CIPHER` (168) and `LINE_PHRASE` (172), as well as the FASTCopy *Administrator's Guide* for details on how to set the encryption parameters in the **Security and Administration Files** on the receiving node.

Attributes

OVERRIDE
POSITIONAL

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LINE_PHRASE

Syntax: `line_phrase=`*`key`*

The `line_phrase` qualifier specifies a key string to be used when encrypting files with the encryption method specified by the `LINE_CIPHER` qualifier. (If no method is specified, FASTCopy assumes the encryption method to be `des-cbc`). Only the data (i.e. the files) are encrypted using this key; in particular, messages sent back and forth between the FASTCopy application and the FASTCopy daemon on the remote node are not encrypted.

By itself, the `line_phrase` qualifier does not cause FASTCopy to encrypt the data; this is achieved through the `line_encrypt` qualifier. The `line_phrase` qualifier only tells FASTCopy what key to use if it is required to encrypt a file.

For the encryption to take place, both the cipher and key phrase specified in the operation's command line must match those specified in the receiving node's **Security and Administration Files**. Otherwise, an error occurs. Thus, you can only use this form of encryption if the cipher and key phrase are known on both the local and the remote node.

The *key* chosen can be any string including upper and lower case characters, numbers, non-alphanumeric keyboard symbols, spaces, etc. If the string includes spaces, it should be enclosed in quotes; the quotes must be escaped with backslashes if used from a UNIX shell. For good cryptographic strength, the string should be at least 10 characters long. When choosing a *key*, exercise the same considerations you would use to choose a good password, i.e. avoid using recognizable names, words or phrases that can be easily guessed.

See Appendix B: Encryption Methods (290), for an explanation of the various encryption methods available.

See `LINE_CIPHER` (168) and `LINE_ENCRYPT` (170), as well as the FASTCopy *Administrator's Guide* for details on how to set the encryption parameters in the **Security and Administration Files** on the receiving node.

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

MAC

Syntax: `mac[=`*`phrase`*`]`

The `mac` qualifier specifies that all messages passed between the local and the remote node should be authenticated using the given key *phrase*. When the `mac` qualifier is used, all the data sent over the line - the transferred files *and* any communication between the FASTCopy application and the remote FASTCopy daemon - is signed with the specified key. This allows both sides to ensure that anything they receive during the operation does indeed come from the expected source, and that no data was modified or replaced in transit by a third party.

The `mac` qualifier can be included both in an operation's command line and in the **Security and Administration Files** of the receiving node. If either of these specifies `mac` without a key *phrase*, FASTCopy will use an internal key to sign all its communication packets during the operation. This protection, however, has no cryptographic strength, since the FASTCopy program can be “cracked” by a third party and the internal key used to forge communication packets.

If the `mac` qualifier is used with a key phrase, the phrase in the operation's command line must be the same as the phrase in the receiving node's **Security and Administration Files**, or the operation will fail. When an agreed-upon key is used, FASTCopy provides both parties with strong cryptographically-protected authentication of their partner's identity.

The *phrase* chosen can be any string including upper and lower case characters, digits, non-alphanumeric keyboard symbols, spaces, etc. If the string includes spaces, it should be enclosed in quotes; the quotes must be escaped with backslashes if used from a UNIX shell. For good cryptographic strength, the *phrase* should be at least 10 characters long. When choosing a key *phrase*, exercise the same considerations you would use to choose a good password, i.e. avoid using recognizable names, words or phrases that can be easily guessed.

See the FASTCopy *Administrator's Guide* for details on how to set message authentication in the receiving node's **Security and Administration Files**.

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

SL_SECURITY

Syntax: `sl_security`

The `sl_security` qualifier specifies that if any of the files or directories in the source file specification has an associated **Security and Administration Files**, this file should be transferred to the target node along with the data file.

Security and Administration Files that are specific to a file or directory contain rules governing file transfer operations that involve that file or directory. These rules are in addition to any rules specified in a node's general **Security and Administration Files** and the **Security and Administration Files** of any higher-level directories. Security files can be transferred only if this is explicitly allowed in the **Security and Administration Files** on both nodes, and in any security files of directories in the target path.

A file's associated security file is called ***filename.extension*.sl_sec** on UNIX, Windows NT and Windows 9x systems, and ***filename.extension*_sl_sec** on OpenVMS systems.
A directory's associated security file is a file inside that directory ending with the suffix **.sl_sec**.

For more about file and directory level **Security and Administration Files**, see Chapter 4 of the FASTCopy *Administrator's Guide*: Security and Administration.

Attributes

OVERRIDE
POSITIONAL

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

General SSL Qualifiers

Note: In order to utilize FASTCopy's SSL feature, you must have a license that supports SSL encryption installed on both the local and the remote nodes.

FASTCopy v2.6 and above supports OpenSSL client-server/server-client encryption and authentication.

From FASTCopy v2.6.6 the MS-CAPI standard is also supported.

The following section contains qualifiers that are relevant for both OpenSSL and MS-CAPI standards. For qualifiers specific to OpenSSL, see OpenSSL Qualifiers (184). For qualifiers specific to MS-CAPI, see MS-CAPI SSL Qualifiers (194).

The following qualifiers are described in this section:

- `SECURITY_PROVIDER` (179)
- `AUTHENTICATE` (180)
- `CERT` (181)
-

- CIPHER (182)
- PEER_COMMON_NAME (183)

SECURITY_PROVIDER

Syntax: `-security_provider=[openssl|ms]`

Default: If no qualifier is specified, FASTCopy will use OpenSSL.

The `–security_provider` qualifier specifies what SSL security provider to use for SSL sessions. If you specify `–security_provider=ms` (MS-CAPI), ensure that you have imported the required certificates and that FASTCopy Daemon has been configured accordingly. Refer to the *FASTCopy SSL Guide* for instructions on how to perform both of these tasks.

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

AUTHENTICATE

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax: `authenticate=[cert|name|all]`

Default: `authenticate=all`

The `authenticate` qualifier can be used with each of the three values described below. If no value is specified, the default value of `all` is chosen.

cert	The client verifies that the server's certificate is valid. If the server's certificate is not valid, the operation fails.
name	The client verifies whether the `Common_Name` field of the Server's certificate matches either the hostname of the server, or the value of the `peer_common_name` qualifier described below.
all	The client verifies both of the above. This is the default.

See also: `PEER_COMMON_NAME` (183)

Attributes

LIST VALUE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Contact support

CERT

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax (OpenSSL): `cert=`*`full_path\certificate_file`*

Syntax (MS-CAPI): `cert=<`*`certificate_subject`*`>`

In OpenSSL, the `cert` qualifier should be used to specify the pathname of the client's certificate.

In MS-CAPI, the `cert` qualifier should be used to specify all or part of the contents of the subject field in the client's certificate.

MS-CAPI Example of Full Subject:

```
-certificate=”C=IL,S=Israel,L=Petach-
Tikva,O=SoftLink,OU=Development,CN=Client-Demo,E=info@softlink.com”
```

You do not have to specify the entire contents of the subject field. As long as the value that you specify is exclusive to that certificate, you can specify part of the contents.

MS-CAPI Example of Partial Subject:

```
-certificate=E=info@softlink.com
```

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX (only OpenSSL)	(Yes)
OpenVMS	Contact support

CIPHER

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax: `cipher=[des|des3|RC2|RC4]`
Default: `cipher=des3`

The `cipher` qualifier encrypts the SSL session and accepts one of the following encoding algorithms:

AES	Advanced Encryption Cipher (Not supported by MS-CAPI)
Des	DES (Data Encryption Standard) applies a 56-bit key to each 64-bit block of data.
Des3	Triple DES
RC2	RC2 (Rivet's Cipher 2) is a variable key-size block cipher.
RC4	RC4 is a variable key-size block cipher with a key size range of 40 to 128 bits. It is faster than DES and is exportable with a key size of 40 bits.

Note: MSCAPI does not support the AES session cipher. Thus, if you are transferring files to or retrieving files from a server that uses MS-CAPI, you must select a cipher that MS-CAPI supports.

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Contact support

PEER_COMMON_NAME

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Qualifier: `peer_common_name=`*`name`*

The `peer_common_name` qualifier indicates the expected contents of the server certificate's **Common_Name** field, when it is not the server's hostname.

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

OpenSSL Qualifiers

The following qualifiers can be used to enable SSL functionality:

- CA_DIR (185)
- CA_FILE (186)
- ENCRYPT (187)
- EXPORT_SCRAMBLED_KEY_PHRASE (188)
- KEY (190)
- KEY PHRASE (191)
- SCRAMBLE_KEY_PHRASE (192)
- SCRAMBLED_KEY_PHRASE (193)

Refer to the *FASTCopy SSL Guide* for a detailed explanation on how to implement FASTCopy's SSL feature.

CA_DIR

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax: `ca_dir=`*`full_path\directory`*

The `ca_dir` qualifier specifies the path to a directory containing Trusted Certificate Authority (CA) certificates. Use this qualifier instead of the `ca_file` qualifier if the client trusts more than one CA.

See also: `CA_FILE` (186)

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Contact support

CA_FILE

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax: `ca_file=`*`full_path\ca_file`*

The `ca_file` qualifier specifies the path and filename of the Trusted Certificate Authority's certificate.

If the client trusts more than one CA, use the `ca_dir` qualifier described above.

See also: `CA_DIR` (185)

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Contact support

ENCRYPT

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax: `encrypt`

The `encrypt` qualifier indicates that the client wishes to establish an SSL session (not necessarily authenticated). The `encrypt` qualifier should not be confused with the `line_encrypt` qualifier.

See also: `LINE_ENCRYPT` (170)

Attributes

NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Contact support

EXPORT_SCRAMBLED_KEY_PHRASE

Note: This qualifier is supported from FASTCopy version 2.6.5 upwards

Syntax: `-export=`*`path_filename.cf`*

This qualifier should only be used as part of the procedure for starting the FASTCopy daemon with the required certificate and key.

For the full procedure, see Appendix E: Starting FASTCopy Daemon with the Required SSL Certificate and Key (318).

The `export` qualifier writes the output of the scramble_key_phrase qualifier to the specified file. The output consists of the pathname of the server's private key, the scrambled key material and the scrambled key phrase. As part of the aforementioned procedure you must delete the path to the file, leaving only the name of the private key file.

Below is an example of a file containing the output of the scramble_key_phrase qualifier after the path to the filename has been deleted, leaving only the name of the private key file.

```
[_cf_header_]
version=2

[KeySection]
Key Name=fc_server_key.pem
Key Material=T6NFFS1P7HGCPKN9B4R8ADU6B1
Scrambled Key Phrase=5J1OJ3TBTS2NFRA3L96FGHDF42
```

See also: SCRAMBLE_KEY_PHRASE (192), SCRAMBLED_KEY_PHRASE (193)

Attributes

NO PARAMETERS

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

KEY

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax: `key=`*`full_path`*`\`*`key_file`*

The `key` qualifier specifies the path and filename of the client's private key file.

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

KEY PHRASE

Note: This qualifier is supported from FASTCopy version 2.6 upwards

Syntax: `key_phrase=`*`passphrase`*

The `key_phrase` qualifier specifies the passphrase for the client's private key file.

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

SCRAMBLE_KEY_PHRASE

Note: This qualifier is supported from FASTCopy version 2.6.5 upwards

Syntax: `scramble_key_phrase`

This qualifier should only be used as part of the procedure for starting the FASTCopy daemon with the required certificate and key.

For the full procedure, see Appendix E: Starting FASTCopy Daemon with the Required SSL Certificate and Key (318).

The `scramble_key_phrase` qualifier scrambles the passphrase to the server's private key file. The scrambled key phrase is split into two parts – the first part is called the *Scambled key phrase* and the second part is called the *Scrambled key material*.

Example:

```
C:\ProgramFiles\SoftLink\fastcopy\bin>fcopy -scramble_key_phrase -key=\"c
:\\programfiles\\softlink\\ssl\\fc_server_key.pem\"
-key_phrase=wellkeptsecret
FASTCopy private key phrase encrypting output :

Scrambled key phrase is 7O1RVSLABMGJQ2H54T6F5T0ME3
Scrambled key material is 4II8EP0B4PD866KE2AVQL449M3
```

Attributes

NEGATABLE
NO PARAMETERS

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

SCRAMBLED_KEY_PHRASE

Note: This qualifier is supported from FASTCopy version 2.6.5 upwards

Syntax: `scrambled_key_phrase`

The `scrambled_key_phrase` qualifier provides added security by instructing FASTCopy to use the scrambled passphrase to the user's private key file instead of the actual passphrase. To be able to use this qualifier you must first scramble your passphrase and store it in the Windows Registry. For further information on this procedure, contact SoftLink Support.

If you do not wish to use the `scrambled_key_phrase` qualifier, you can specify the actual passphrase using the `KEY_PHRASE` (191) qualifier.

See also: `SCRAMBLE_KEY_PHRASE` (192), `KEY_PHRASE` (191).

Attributes

NEGATABLE
NO PARAMETERS

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

MS-CAPI SSL Qualifiers

This section contains qualifiers for securing FASTCopy sessions using the MS-CAPI SSL mechanism.

IMPORTANT: If you want to use MS-CAPI you *must* include the qualifier `-service_provider=ms` in your FASTCopy command line.

See `SECURITY_PROVIDER` (179).

MS-CAPI qualifiers are as follows:

- `CERT_STORE_NAME` (195)
- `CERT_STORE_TYPE` (196)
- `CSP` (197)
- `MS_PROVIDER_TYPE` (198)
- `KEY_CONTAINER` (199)
- `CRL` (200)

Note: It is strongly recommended to read the *FASTCopy SSL Guide* before using these qualifiers. The guide includes instructions on how to import MS-CAPI compliant certificates on the client and server machines. The guide also explains how to configure FASTCopy Daemon to load the server certificate.

CERT_STORE_NAME

Syntax: `-cert_store_name=<`*`certificate store name`*`>`

The `–cert_store_name` qualifier specifies the name of the certificate store to which you imported the client MS-CAPI compliant certificate. For example, if you imported your certificate to the **Personal** store, the qualifier would be written:

```
-cert_store_name=my
```

This is also the default if the `-cert_store_name` qualifier is not specified.

The following is a list of other possible `<`*`certificate store name`*`>` values:

- `ACRS`
- `AddressBook`
- `AuthRoot`
- `CA`
- `Disallowed`
- `REQUEST`
- `Root`
- `Trust`
- `TrustedPeople`
- `TrustedPublisher`
- `UserDS`

Supported Operating Systems

Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

CERT_STORE_TYPE

Syntax: `CERT_STORE_TYPE=[user|machine|enterprise]`

Default: If you do not include the qualifier FASTCopy assumes that you imported the certificate to the current user store (`user`)

The `–cert_store_type` qualifier specifies the certificate store type where:

- `user` is the current user store
- `machine` is the local machine store
- `enteprise` is the local enterprise store

Supported Operating Systems

Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

CSP

Syntax: `-csp=<Cryptographic Service Provider>`

The `-csp` qualifier can be used to specify a Cryptographic Service Provider.

Example:

```
-csp="Microsoft Enhanced DSS and Diffie-Hellman Cryptographic Provider"
```

If the qualifier is not specified, FASTCopy will use the default CSP on your machine.

Supported Operating Systems

Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

MS_PROVIDER_TYPE

Syntax: `-ms_provider_type=<`*`provider_type`*`>`

If necessary, the `-ms_provider_type` qualifier can be used to specify what type of Cryptographic Service Provider you are using.

Supported Operating Systems

Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

KEY_CONTAINER

Syntax: `key_container=<key container name>`

The `-key_container` qualifier can be used to specify a key container name. If the `-key_container` qualifier is not specified, FASTCopy will use your CSP's default key container.

See also: CSP (197)

Supported Operating Systems

Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

CRL

Syntax: `-crl=<`*`crl_pathname`*`>`

The `-crl` qualifier specifies the pathname of a file containing a list of certificates that have been revoked by the CA that issued them.

The file must be created using a propriety utility.

Supported Operating Systems

Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

PIN_REQUIRED

Syntax: `-pin_required`

The `–pin_required` qualifier must be used if you are using a smart card or hardware token that requires you to interactively insert a PIN code.

Supported Operating Systems

Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

Scheduling and Batch Qualifiers

This section details qualifiers that affect the execution of FASTCopy operations under batch schedulers. These include qualifiers for scheduling the operation's execution time and for affecting how it is handled by the Scheduler application. FASTCopy operations are handled differently on each supported platform, so support for these qualifiers varies across platforms. On OpenVMS, scheduling is handled by the internal OpenVMS batch queues.

Scheduling and Batch qualifiers include:

- ASSUME (203)
- BATCH (204)
- CONTROLLER_DOMAIN (206)
- CONTROLLER_PASSWORD (207)
- CONTROLLER_USER (208)
- DAILY (209)
- EVERY (210)
- GROUP (213)
- HOLD_UNTIL (214)
- IGNORE_CWD_ERROR (215)
- JOB_NAME (216)
- ONCE_TIME (217)
- RUN_OPTION (219)
- TERMINATE_AFTER (221)
- TRIGGER_FILE (223)
- WARNING_STATUS (224)

ASSUME

Syntax: `assume=`*`reply`*

The `assume` qualifier issues an automatic reply for queries that FASTCopy issues during a batch operation. Queries can be one or both of the following:

- FASTCopy requesting permission to perform the operation.
- FASTCopy requesting permission to overwrite existing files on the target node.

The *`reply`* value can be one of the following:

Reply Value	Definition
y	Yes
n	No
q	Quit
a	All

When you use FASTCopy in batch mode, and do not specify the `assume` qualifier, *y* is assumed for `CONFIRM` queries and *n* is assumed for `REPLACE` queries.

See also: `CONFIRM` (63), `REPLACE` (87), `SILENT` (53)

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

BATCH

Note: On Windows platforms, this qualifier is supported from FASTCopy version 2.6 upwards.

UNIX/Windows Syntax: `batch`

OpenVMS Syntax: `batch[=`*`batch_queue`*`]`

The `batch` qualifier submits a FASTCopy operation for batch execution. This allows the scheduling and automatic recovery of FASTCopy transfers.

- **On UNIX**, batch execution occurs under the *flogicd* daemon. Once the daemon accepts the operation, it is given a job number and control is returned to the user. The batch daemon is responsible for checking completion status and issuing recovery attempts. To check on the progress of a FASTCopy batch operation, issue the **fmonitor** command with the job number that was assigned to the operation.

- **On OpenVMS**, batch is implemented under the VMS batch queues. A value can be specified with the batch qualifier, specifying the name of a particular *batch_queue* which will handle the operation. The default queue used is SYS$BATCH. Once the queue accepts the operation, it is given a job number and control is returned to the user. The batch queue is responsible for checking completion status and issuing recovery attempts.

- **On Windows**, batch mode is facilitated through the use of native Windows Services (SoftLink Services). By default, the batch job is submitted for immediate execution. Invoking the `run_option` qualifier with the desired value controls the execution time.

IMPORTANT: On Windows, The `batch` qualifier is ineffective unless used in conjunction with the `controller_user` and `controller_password` qualifiers.

Refer to the FASTCopy *User's Guide* for further details about Batch operations.

See also: `RUN_OPTION` (219), `CONTROLLER_USER` (208) and `CONTROLLER_PASSWORD` (207).

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

CONTROLLER_DOMAIN

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `controller_domain=`*`domain_name`*

If the user account belongs to a Windows NT/2000 domain, specify the *`domain_name`*. If the user account is local, omit this qualifier.

See also: `BATCH` (204), `CONTROLLER_USER` (208) and `CONTROLLER_PASSWORD` (207).

Attributes

BATCH

Supported Operating Systems

Windows 9x	No
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

CONTROLLER_PASSWORD

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `controller_password=`*`password`*

The `controller_password` qualifier specifies the correct password for the user account under which the job is submitted. It must be used in conjunction with the `batch` and `controller_user` qualifiers.

See also: `BATCH` (204) and `CONTROLLER_USER` (208)

Attributes

BATCH

Supported Operating Systems

Windows 9x	No
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

CONTROLLER_USER

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `controller_user=`*`user_name`*

The `controller_user` qualifier specifies the user under which the job is submitted. The privileges of the user submitting the job must grant the user "read access" to the files being transferred. It must be used in conjunction with the `batch` and `controller_password` qualifiers.

See also: `BATCH` (204) and `CONTROLLER_PASSWORD` (207).

Attributes

BATCH

Supported Operating Systems

Windows 9x	No
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

DAILY

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `daily=`*`hhmm`*

hh = hour of day (24-hour format)
mm = minutes

The `daily` qualifier specifies the time of day for the execution of a daily batch job. It must be used in conjunction with `run_option=daily`.

See also: `BATCH` (204) and `RUN_OPTION` (219).

Attributes

BATCH

Supported Operating Systems

Windows 9x	No
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

EVERY

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `every=\”`*`number time_period`*`\”`

The `every` qualifier specifies the time period that should elapse between successive executions of a FASTCopy batch operation. It must be used in conjunction with `run_option=every`.

The `time_period` value is one of the following:

- seconds
- minutes
- hours
- days
- weeks

Note: The minimum time period permitted between jobs is 1 minute.

The expression must be enclosed within quotation marks (“”), and each quotation mark must be preceded by a backslash.

For example, to specify a two minute time period:

```
run_option=every every=\”2 minutes\”
```

See also: `BATCH` (204) and `RUN_OPTION` (219).

Attributes

BATCH

Supported Operating Systems

Windows 9x	No
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

FSUBMIT_QUALS

Syntax: `fsubmit_quals="qual1=value qual2=value..."`

The `fsubmit_quals` qualifier is used when a FASTCopy operation is submitted to the *flogicd* batch scheduler. Submitting a batch operation on UNIX is done by calling an external program called *fsubmit*, which submits the operation to *flogicd*'s internal queue. If you are familiar with *fsubmit*, you can use `fsubmit_quals` to modify how *fsubmit* handles the submission of the FASTCopy batch operation.

Refer to the *FASTLogic User's Guide* and the *FASTLogic Reference Guide* for more information on *fsubmit*.

See also: `BATCH` (204) and `RUN_OPTION` (219).

Attributes

BATCH

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

GROUP

Syntax: `group=`*`group_name`*

The `group` qualifier is used to identify a specific logical group as the owner of a batch operation, for the purposes of administration, security and monitoring. The *`group`*_name is an internal FASTCopy identifier and is no way related to the UNIX “Group” concept.

Specific users can be given the privilege to monitor or alter a job based on the job’s group identifier.

Groups should be defined in the **Security and Administration Files**.

See also: `BATCH` (204).

Attributes

BATCH

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

HOLD_UNTIL

Syntax: `hold_until=`*`time`*

The `hold_until` qualifier instructs FASTCopy's Scheduler to hold the execution of the operation until the time specified.

For information on what constitutes a valid time frame expression, see Appendix A: Setting Time and Time-Frame Expressions (285).

See also: `BATCH` (204).

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

IGNORE_CWD_ERROR

Note: This qualifier is supported from FASTCopy version 2.6.5 upwards.

Syntax: `ignore_cwd_error`

The `ignore_cwd_error` qualifier prevents FASTCopy from changing the current working directory to the one written in its context file. This qualifier is should only be used if a batch command is run as a pre/post/exit command of another FASTCopy batch job.

Attributes

BATCH
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Contact support

JOB_NAME

`job_name=`*`name`*

The `job_name` qualifier submits a FASTCopy operation for batch execution under a particular name. This qualifier can only be used on OpenVMS platforms, where FASTCopy batch operations are implemented under the VMS batch queues.

See also: `BATCH` (204).

Attributes

BATCH

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	No
OpenVMS	Yes

ONCE_TIME

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `once_time=`*`DDMMYYYYhhmm`*

DD = day
MM = month
YYYY = year
hh = hour of day (24-hour format)
mm = minutes

The `once_time` qualifier specifies a date and time for the once-only execution of a FASTCopy batch operation. It must be used in conjunction with `run_option=once`.

Example

Moe wants to transfer the file **a.txt** from his local machine to the remote node **NT13**. He wishes the transfer to occur on April 10th, 2002 at 1 p.m.To accomplish this, Moe issues the following command:

WINDOWS

```
fcopy a.txt \\NT13\c:\temp -controller_user=charlie
-controller_password=charlie –batch –run_option=once
once_time=100420021300 –user=Jim –pass=Gully
```

See also: `BATCH` (204) and `RUN_OPTION` (219).

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

RUN_OPTION

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `run_option=[immediate|once|every|daily|triggerfile]`
Default: `run_option=immediate`

The `run_option` qualifier schedules the execution time frame for a batch operation. The default value is immediate. If `run_option` is not specified, the batch operation runs immediately.

Otherwise, use one of the following values:

Value	Description
Once	Runs the batch operation once, at the scheduled time. Must be used with the `ONCE_TIME` (217) qualifier.
Every	Runs the batch operation every x seconds, minutes, hours, days or weeks. Must be used with the `EVERY` (210) qualifier.
Daily	Runs a daily batch operation at the specified time. Must be used with the `DAILY` (209) qualifier.
Triggerfile	Runs the batch operation only after the specified file has been created in, or copied to the designated location. Must be used with the `TRIGGER_FILE` (223) qualifier.

Each of the `run_option` values listed above must be used with its corresponding qualifier.

See also: `BATCH` (204), `ONCE_TIME` (217), `EVERY` (210), `DAILY` (209) and `TRIGGER_FILE` (223).

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

TERMINATE_AFTER

Syntax: `terminate_after=`*`time`*

The `terminate_after` qualifier terminates an operation run by the Scheduler if it is not completed by the specified time. If the time arrives and the operation is still running, it will be aborted. If the operation is between recovery attempts, it will be stopped and the next recovery directive will not be issued.

An operation terminated for this reason generates an error.

The `time_frame_expression` value is specified in one of the following formats:

- hhmm - time to use with current date
- DDMMYY
- DDMMYYYY
- DDMMYYYYhhmm
- DDMMYYYYhhmmss

Syntax	DD	MM	YY	YYYY	hh	mm	ss
Value	day	month	year (2 digits)	year (4 digits)	hour of day (24-hour format)	minutes	seconds

For information on what constitutes a valid time frame expression, see Appendix A: Setting Time and Time-Frame Expressions (285).

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

TRIGGER_FILE

Note: This qualifier is supported from FASTCopy version 2.6 upwards.

Syntax: `trigger_file=`*`full_path`*`\`*`filename`*

The `trigger_file` qualifier specifies the name of a trigger file (filename) and the location (full_path) in which the trigger file must be created or copied to, for the batch operation to start.

It must be used in conjunction with `run_option=`*`triggerfile`*`.`

See also: `BATCH` (204) and `RUN_OPTION` (219).

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

WARNING_STATUS

Syntax: `warning_status=error|warning|success`

If a FASTCopy operation terminates with a warning, the `warning_status` qualifier specifies the exit status that it should return to the calling program. By default, FASTCopy exits with a warning. However, some programs can cope only with an error or success status.

Using the `warning_status` qualifier, you can tell FASTCopy to return an *error* or *success* status when it exits with a warning.

See also: `BATCH` (204).

Attributes

OVERRIDE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

Execution Qualifiers

FASTCopy's post and pre transfer processing features include qualifiers that allow task activation linked to different stages of a FASTCopy operation. The tasks can also be scheduled and recovered using FASTCopy's scheduling mechanisms.

Execution qualifiers are as follows:

- EXECUTE_ONLY (227)
- EXIT_COMMAND (229)
- EXIT_LOG_FILE (230)
- EXIT_PARAMETER (231)
- EXIT_SYNC (233)
- INIT_COMMAND (234)
- INIT_LOG_FILE (235)
- INIT_PARAMETER (236)
- INIT_SYNC (237)
- LOCAL_COMMAND (238)
- LOCAL_LOG_FILE (240)
- LOCAL_PARAM (241)
- LOCAL_QUEUE (244)
- LOCAL_SYNC (245)
- LOCAL_PRE_COMMAND (246)
- LOCAL_PRE_LOG_FILE (247)
- LOCAL_PRE_PARAM (248)
- LOCAL_PRE_QUEUE (249)
- LOCAL_PRE_SYNC (250)
- POST_INDEPENDANT_COMMANDS (251)
- POST_TRANSFER_ORDER (252)
- PRE_INDEPENDANT_COMMANDS (253)
- PRE_TRANSFER_ORDER (254)
- REMOTE_COMMAND (255)

- REMOTE_LOG_FILE (257)
- REMOTE_PARAM (258)
- REMOTE_QUEUE (261)
- REMOTE_SYNC (262)
- REMOTE_PRE_COMMAND (263)
- REMOTE_PRE_LOG_FILE (264)
- REMOTE_PRE_PARAM (265)
- REMOTE_PRE_QUEUE (266)
- REMOTE_PRE_SYNC (267)
- RETRIEVE_REMOTE_LOG (268)

EXECUTE_ONLY

Syntax: `execute_only[=`*`remote_node`*`]`

The `execute_only` qualifier lets you activate tasks using the `remote_command` and `local_command` qualifiers independent of any file transfer operation. When `execute_only` is specified, FASTCopy activates the `remote_command` on the remote_node that is specified or the `local_command` on the local node. When a remote_node is not specified, only the `local_command` can be executed.

Specifying `execute_only` with the `remote_command`, `batch` and `hold_until` qualifiers provides guaranteed and scheduled execution of commands on remote nodes.

See also: `LOCAL_COMMAND` (238), `REMOTE_COMMAND` (255)

Example 1

Executing a local command through FASTCopy without performing a file transfer:

The following FASTCopy command line instructs FASTCopy to carry out an external command without performing a file transfer (by using the `execute_only` qualifier). Since no remote node was specified, only a `local_command` can be executed. In the example, FASTCopy executes a `local_command`, calling the UNIX program *date* and outputs its result to the file **/tmp/date.log** on the local node **Sun**. The **cat** command is used to see the contents of the output file, verifying that the `local_command` has been executed.

UNIX

```
>fcopy -execute_only -local_command=\"date,/tmp/date.log\"
FCOPY-I-L_COMMAND - Local command initiated
>cat /tmp/date.log
Sun Dec 31 18:12:27 GMT+0200 2001
>
```

Example 2

Executing a remote command using command parameters without performing a file transfer:

The example below is only slightly more complicated than the previous one. Here, the user specifies the remote node **alf** as the node on which he/she wishes the `execute_only` operation to be performed. The command **uname** is specified as the `remote_command`, with the parameter **-a**. The result of the command is outputted to the file **/tmp/uname.log** on the remote node **alf**. The second command line checks the contents of that file on the **alf** node to confirm the successful execution of the remote command.

UNIX

```
>fcopy -execute_only=alf -remote_command=\"uname,/tmp/uname.log\"
-remote_param=\"-a\"
FCOPY-I-R_COMMAND - Remote command initiated

>rsh alf cat /tmp/uname.log
OSF1 alf V2.0 240 alpha
```

Attributes

NO_PARAMETERS

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

EXIT_COMMAND

Syntax: `exit_command=`*`command`*

The `exit_command` qualifier specifies a command to be executed upon the termination of a FASTCopy operation run in batch mode.

The termination may either be a successful termination or any kind of premature termination of the FASTCopy operation. The command is executed on the local node (the node that issued the FASTCopy command).

The `exit_command` qualifier can only be used with the batch qualifiers. A typical use is to provide an asynchronous notification of a FASTCopy operation's state to an external application.

Parameters can be passed to the command with the `exit_parameter` qualifier, and the output of the command can be directed to a file using the `exit_log_file` qualifier.

See also: `EXIT_PARAMETER` (231), `EXIT_LOG_FILE` (230), `EXIT_SYNC` (233), `INIT_COMMAND` (234), `LINK_DOWN_COMMAND` (270) and `LINK_UP_COMMAND` (272).

Attributes

OVERRIDE
LIST_VALUE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

EXIT_LOG_FILE

Syntax: `exit_log_file=`*`file`*

The `exit_log_file` qualifier specifies an output file to which the output of the command specified with the `exit_command` qualifier should be redirected. The `exit_command` qualifier is issued locally after the FASTCopy operation is completed.

The `exit_log_file` qualifier can only be used with the `exit_command` qualifier.

See also: `EXIT_COMMAND` (229), `EXIT_PARAMETER` (231) and `EXIT_SYNC` (233).

Attributes

OVERRIDE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

EXIT_PARAMETER

Syntax: `exit_parameter=`*`parameter`*

The `exit_parameter` qualifier passes a parameter to the command specified by the `exit_command` qualifier after it is activated.

The parameter is a string of characters that can include variables. The variables are interpreted by the *flogicd* daemon prior to the activation of the command specified by the `exit_command` qualifier.

Valid variables for the `exit_parameter` qualifier are as follows:

%S - exit status string	Possible values are: ▪ "ERROR" ▪ "SUCCESS"
%s - exit status value	Possible values are: ▪ 0 for error ▪ 1 for success
%D - detailed status string (reason for termination)	Possible values are: ▪ "UNEXPECTED" ▪ "RETRY LIMIT" ▪ "FCOPY DONE"
%d - detailed status value	Possible values are: ▪ 0 for unexpected. ▪ 1 for retry limit. ▪ 2 for done.

See also: `EXIT_COMMAND` (229), `LINK_DOWN_COMMAND` (270), `LINK_UP_COMMAND` (272), `LINK_DOWN_PARAMETER` (271), `LINK_UP_PARAMETER` (273) and `BATCH` (204).

Attributes

OVERRIDE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

EXIT_SYNC

Syntax: `exit_sync`

The `exit_sync` qualifier specifies that FASTCopy should wait for the exit command to be completed before the operation ends; by default, FASTCopy considers the exit command successful if it was successfully issued. When the `exit_sync` qualifier is used, the command is considered successful - and the operation can end - only if it returns with a status of 0. Otherwise, FASTCopy attempts recovery, issuing the command again. This qualifier is not supported on OpenVMS platforms, where FASTCopy's post-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `EXIT_COMMAND` (229), `EXIT_PARAMETER` (231), and `EXIT_LOG_FILE` (230).

Attributes

OVERRIDE
NEGATABLE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

INIT_COMMAND

Syntax: `init_command=command`

The `init_command` qualifier specifies a command to be executed locally before the FASTCopy operation begins.

The `init_command` qualifier can only be used with the batch qualifiers. A typical use is to turn on a dial-up connection required for the transfer.

Parameters can be passed to the command using the `init_parameter` qualifier.

The output of the command can be directed to a file using the `init_log_file` qualifier.

See also: `INIT_LOG_FILE` (235), `INIT_PARAMETER` (236), `INIT_SYNC` (237) and Scheduling and Batch Qualifiers (202).

Attributes

OVERRIDE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

INIT_LOG_FILE

Syntax: `init_log_file=`*`file`*

The `init_log_file` qualifier specifies an output file to which the output of the `init_command` qualifier should be redirected. The `init_command` qualifier is issued locally before the FASTCopy operation begins.

The `init_log_file` qualifier can only be used with the `init_command` qualifier.

See also: `INIT_COMMAND` (234), `INIT_PARAMETER` (236), and `INIT_SYNC` (237).

Attributes

OVERRIDE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

INIT_PARAMETER

Syntax: `init_parameter=`*`param`*

-Or-

`init_paramter=”`param1 param2… "

When an `init_command` is activated, the parameter specified with `init_parameter` is passed to the command after it is activated.

The parameter consists of a string of characters passed as command line arguments to the activated program. To specify multiple parameters, separate the parameters from each other by spaces and enclose with quotation marks. If used from a UNIX shell, the quotes should be escaped using backslashes.

See also: `INIT_COMMAND` (234) and `INIT_SYNC` (237).

Attributes

OVERRIDE
LIST_VALUE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

INIT_SYNC

Syntax: `init_sync`

The `init_sync` qualifier specifies that FASTCopy should wait for the command specified with the `init_command` qualifier to be completed before starting the operation; by default, FASTCopy considers the command successful if it was successfully issued. When `init_sync` is used, the command is considered successful only if it returns with a status of 0. Otherwise, FASTCopy attempts recovery, issuing the command again. This qualifier is not supported on OpenVMS platforms, where FASTCopy's pre-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `INIT_COMMAND` (234), `INIT_PARAMETER` (236), and `INIT_LOG_FILE` (235).

Attributes

OVERRIDE
NEGATABLE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

LOCAL_COMMAND

Syntax: `local_command=`*`command`*

The `local_command` qualifier executes a command on the local node (i.e. the node that issues the FASTCopy command), following the successful completion of the file transfer. The FASTCopy operation is completed only after this command is successfully activated.

On UNIX, the command value can be a system command or an executable script. On OpenVMS, the command value must specify a command file (ending in .com). On Windows NT and Windows 9x/2000/XP, the command value must be the name of an executable file (ending in **.bat** or **.exe**)

You can pass parameters to the command with the `local_param` qualifier. You can also specify an output file with the `local_log_file` qualifier.

See also: `EXECUTE_ONLY` (227), `LOCAL_PARAM` (241), `LOCAL_LOG_FILE` (240), `LOCAL_SYNC` (245) and `REMOTE_COMMAND` (255).

Examples

The following FASTCopy command requests the activation of the *date* command after successful transfer of the file **a.a** to the remote node **hipi**. Since no output options were specified, the output of the `local_command` will be sent to the standard output.

The FASTCopy command output will be:

UNIX

```
/usr/develop> fcopy a.a hipi:/tmp/b.b -local_command=date

FCOPY-S-FILE_COPIED - File '/usr/develop/a.a' FASTCopied to file
'hipi:/tmp/b.b' (11 Bytes)
FCOPY-I-L_COMMAND - Local command initiated
```

This is followed by the output of the `local_command` qualifier that is executed asynchronously with the FASTCopy command:

UNIX

```
Sun Jan 14 10:20:44 IST 2002
```

Attributes

OVERRIDE
LIST_VALUE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	yes

LOCAL_LOG_FILE

Syntax: `local_log_file=`*`file`*

The `local_log_file` qualifier specifies an output file to which the output of the `local_command` should be redirected. The `local_command` is issued locally after the transfer is completed.

The `local_log_file` qualifier can only be used with the `local_command` qualifier.

See also: `EXECUTE_ONLY` (227), `LOCAL_COMMAND` (238), `LOCAL_PARAM` (241), and `LOCAL_SYNC` (245).

Example

The following command redirects the output of the *date* command (in the `local_command` example) to the file **/tmp/date.log**.

UNIX

```
> fcopy a.a hipi:/tmp/b.b -local_command=date
-local_log_file=/tmp/date.log
```

Attributes

OVERRIDE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

LOCAL_PARAM

Syntax: `local_param=`*`parameter_value`*

When the command specified by the `local_command` qualifier is activated, the parameter specified with `local_param` is passed to the command on the local node.

The *`parameter_value`* is a string of characters that can include variables, which are interpreted by FASTCopy prior to the activation of the *command* specified by the `local_command` qualifier.

If the parameter_value contains spaces it must be enclosed in quotation marks.

Valid parameter_value variables for the `local_param` qualifier are as follows:

%s Source filename of the last file that was transferred.

%t Target filename of the last file that was transferred.

%f Last file that was transferred; With local_param, this is converted to the local filename.

%1 Converts the succeeding filename to the format of the system on which the local_command or remote_command is executed, rather than being translated on the local machine. This is a very useful modifier during cross-platform operations.

See also: `EXECUTE_ONLY` (227), `LOCAL_COMMAND` (238), `LOCAL_SYNC` (245), `LOCAL_LOG_FILE` (240), and `REMOTE_PARAM` (258).

Note: From version 2.3.3 and higher, this qualifier replaces the `p_local_command` qualifier from earlier versions, which is now deprecated.

Examples

Assume that you are transferring all the files matching **a.*** from your current working directory, the contents of which are listed below:

UNIX

```
/usr/develop/yossi/fastcopy/src> ls -l a.*
-rw-r.r.   1 root   other            5 Jan 21 16:02 a.a
-rw-r.r.   1 root   other         1074 Sep 11 15:36 a.log
```

The following FASTCopy command will transfer both files and issue a `local_command` that will relate to the last file transferred. Specifying a variable as the `local_param` makes it possible to issue the command without knowing at the beginning of the operation which of the files will be transferred last.

UNIX

```
/usr/develop/yossi/fastcopy/src> fcopy .a.*. hipi:/tmp -replace
-local_command=\”ls,/tmp/ls .log\” -local_param=\”-l %s\”
FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/fastcopy/src/a.a'
FASTCopied to file 'hipi:/tmp/a.a' ( 5 Bytes )
FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/fastcopy/src/a.log'
FASTCopied to file 'hipi:/tmp/a.log' ( 1074 Bytes )
FCOPY-I-L_COMMAND - Local command initiated
```

When you type the output file of the `local_command`, you can see that the variable in the `local_param` was converted to the name of the last transferred file and contains the output of the *ls* command for that file.

UNIX

```
/usr/develop/yossi/fastcopy/src> cat /tmp/ls.log
-rw-r.r.   1 root     other       1074 Sep 11 15:36
/usr/develop/yossi/fastcopy/src>
```

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LOCAL_QUEUE

Syntax: `local_queue=queue`

The `local_queue` qualifier submits a FASTCopy local command for execution under the named queue rather than the default OpenVMS batch queue. This qualifier can only be used on OpenVMS platforms, where FASTCopy's post-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `EXECUTE_ONLY` (227), `LOCAL_COMMAND` (238), and `LOCAL_PARAM` (241).

Attributes

BATCH
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	No
OpenVMS	Yes

LOCAL_SYNC

Syntax: `local_sync`

The `local_sync` qualifier specifies that FASTCopy should wait for a local command to be completed before ending the operation; by default, FASTCopy considers a local command successful if it was successfully issued. When `local_sync` is used, the local command is considered successful only if it returns with a status of 0. Otherwise, FASTCopy attempts recovery, issuing the command again. This qualifier is not supported on OpenVMS platforms, where FASTCopy's post-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `EXECUTE_ONLY` (227), `LOCAL_COMMAND` (238), and `LOCAL_PARAM` (241).

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

LOCAL_PRE_COMMAND

Syntax: `local_pre_command=`*`command`*

The `local_pre_command` qualifier triggers a command on the local node (the node that issues the FASTCopy command), before the start of the file transfer operation. By default, the FASTCopy operation continues only after this command is successfully completed. To enable the FASTCopy operation to continue before the local pre command is completed, use the negated form of the `LOCAL_PRE_SYNC` qualifier (`nolocal_pre_sync`).

On UNIX, the command value can be a system command or an executable script. On OpenVMS, the command value must specify a command file (ending in **.com**). On Windows NT and Windows 9x/2000/XP, the *command* value must be the name of an executable file (ending in **.bat** or **.exe**).

You can pass parameters to the command with the `local_pre_param` qualifier. You can also designate an output file with the `local_pre_log_file` qualifier.

See also: `EXECUTE_ONLY` (227), `LOCAL_PRE_PARAM` (248), `LOCAL_PRE_LOG_FILE` (247), `LOCAL_PRE_SYNC` (250), and `REMOTE_PRE_COMMAND` (263).

Attributes

OVERRIDE
LIST_VALUE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LOCAL_PRE_LOG_FILE

Syntax: `local_pre_log_file=`*`file`*

The `local_pre_log_file` qualifier specifies an output file to which the output of the local pre-transfer command should be redirected.
The `local_pre_command` is issued locally before transfer begins.

The `local_pre_log_file` qualifier can only be used with the `local_pre_command` qualifier.

See also: `EXECUTE_ONLY` (227), `LOCAL_PRE_COMMAND` (246), `LOCAL_PRE_PARAM` (248) and `LOCAL_PRE_SYNC` (250).

Attributes

OVERRIDE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

LOCAL_PRE_PARAM

Syntax: `local_pre_param=`*`parameter_value`*

When a `local_pre_command` is activated, the parameter_value specified by the `local_pre_param` qualifier is passed to the command on the local node.

The parameter_value is a string of characters that can include variables that are interpreted by FASTCopy prior to the activation of the `local_pre_command`.

If the parameter_value contains spaces it must be enclosed in quotes.

See also: `EXECUTE_ONLY` (227), `LOCAL_PRE_COMMAND` (246), `LOCAL_PRE_LOG_FILE` (247), `LOCAL_PRE_SYNC` (250) and `REMOTE_PRE_PARAM` (265).

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

LOCAL_PRE_QUEUE

Syntax: `local_pre_queue=queue`

The `local_pre_queue` qualifier submits a FASTCopy local pre-transfer command for execution under the named queue rather than the default OpenVMS batch queue.

This qualifier can only be used on OpenVMS platforms, where FASTCopy's pre-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `EXECUTE_ONLY` (227), `LOCAL_PRE_COMMAND` (246) and `LOCAL_PRE_PARAM` (248).

Attributes

BATCH
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	No
OpenVMS	Yes

LOCAL_PRE_SYNC

Syntax: `local_pre_sync`

The `local_pre_sync` qualifier specifies that FASTCopy should wait for a local command to be completed before beginning the transfer stage. By default, FASTCopy considers a local pre-transfer command successful if it was successfully issued. When the `local_pre_sync` qualifier is used, the local pre-transfer command is considered successful only if it returns a status of 0. Otherwise, FASTCopy attempts recovery, issuing the command again.

This qualifier is not supported on OpenVMS platforms, where FASTCopy's pre-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `EXECUTE_ONLY` (227), `LOCAL_PRE_COMMAND` (246) and `LOCAL_PRE_PARAM` (248).

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

POST_INDEPENDANT_COMMANDS

Syntax: `post_independent_commands`

The `post_independent_commands` qualifier is relevant only when the operation includes both local and remote post transfer commands. It specifies that if the operation is aborted after only one of the commands was triggered successfully, only the remaining command should be carried out during recovery. This is the default situation. The alternative, when the qualifier is negated, forces FASTCopy to activate both commands whenever it attempts to recover.

This qualifier is not supported on OpenVMS platforms, where FASTCopy's post-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `LOCAL_COMMAND` (238) and `REMOTE_COMMAND` (255).

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

POST_TRANSFER_ORDER

Syntax: `post_transfer_order=local_before_remote|remote_before_local|parallel`

The `post_transfer_order` qualifier is relevant only when the operation includes both local and remote post transfer commands. It specifies in what order these commands should be triggered.

The qualifier must be used with one of three possible values:

local_before_remote Triggers the remote command only after the local command is activated or, if the `local_sync` qualifier is used, after it is completed.

remote_before_local Triggers the local command only after the remote command is activated or, if the `remote_sync` qualifier is used, after it is completed

parallel Triggers both commands together (in random order).

This qualifier is not supported on OpenVMS platforms, where FASTCopy's post-transfer processing commands are implemented under the OpenVMS batch queues.

Note: This qualifier is only supported from version 2.3.4 and higher.

See also: `LOCAL_COMMAND` (238) and `REMOTE_COMMAND` (255)

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

PRE_INDEPENDANT_COMMANDS

Syntax: `pre_independent_commands`

The `pre_independent_commands` qualifier is relevant only when the operation includes both local and remote pre transfer commands. It specifies that if the operation is aborted after only one of the commands was triggered successfully, only the remaining command should be carried out during recovery. This is the default situation. The alternative, when the qualifier is negated, forces FASTCopy to activate both commands whenever it attempts to recover.

This qualifier is not supported on OpenVMS platforms, where FASTCopy's pre-transfer processing commands are implemented under the OpenVMS batch queues.

Note: This qualifier is only supported from version 2.3.4 and higher.

See also: `LOCAL_PRE_COMMAND` (246) and `REMOTE_PRE_COMMAND` (263)

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

PRE_TRANSFER_ORDER

Syntax: `pre_transfer_order=local_before_remote|`
`remote_before_local|parallel`

The `pre_transfer_order` qualifier is relevant only when the operation includes both local and remote pre transfer commands. It specifies in what order these commands should be triggered.

The qualifier must be used with one of three possible values:

local_before_remote Triggers the remote command only after the local command is activated or, if the local_sync qualifier is used, after it completes.

remote_before_local Triggers the local command only after the remote command is activated or, if the remote_sync qualifier is used, after it completes.

parallel Triggers both commands together (in random order).

This qualifier is not supported on OpenVMS platforms, where FASTCopy's post-transfer processing commands are implemented under the OpenVMS batch queues.

Note: This qualifier is only supported from version 2.3.4 and higher.

See also: `LOCAL_PRE_COMMAND` (246) and `REMOTE_PRE_COMMAND` (263)

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

REMOTE_COMMAND

Syntax: `remote_command=`*`command`*

The `remote_command` qualifier executes a command on the remote node, following successful completion of a file transfer. The FASTCopy operation is completed only after the successful activation of this command.

On UNIX, the command value can be a system command or an executable script. On OpenVMS, the command value must specify a command file (ending in **.com**). On Windows NT and Windows 9x/2000/XP, the command value must be the name of an executable file (ending in **.bat** or **.exe**).

You can specify an output file to which the command's output will be redirected using the `remote_log_file` qualifier. The root directory for the file is the home directory of the user under which you logged in on the remote node.

Parameters can be passed to the command with the `remote_param` qualifier. The remote command is executed with the permissions of the user you specify when connecting to the remote node, but programs or scripts do not have access to any of that user's environment variables. These are set only when an actual login takes place.

See also: `EXECUTE_ONLY` (227), `REMOTE_PARAM` (258), `REMOTE_LOG_FILE` (257), `REMOTE_SYNC` (262) and `REMOTE_QUEUE` (261).

Examples

In the following example, FASTCopy:

- Transfers a file from the **SUN** node (the local node) to an **HP** node.
- Activates the *uname* command on the remote node after the transfer.

Local node *uname* output:

UNIX

```
/usr/develop/yossi> uname -a
SunOS sun2 5.3 Generic sun4m sparc
```

The FASTCopy command with remote command, log file and parameter:

UNIX

```
/usr/develop/yossi> fcopy a.a hipi:/tmp -replace
-remote_command=\”uname\” -remote_log_file=\”/tmp/uname .log\”
-remote_param=\”-a\”
FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/a.a' FASTCopied to
file 'hipi:/tmp/a.a' ( 12 Bytes )
FCOPY-I-R_COMMAND - Remote command initiated
```

The contents of the output file specified for the remote command on the **HP** node (**/tmp/uname.log**) are:

UNIX

```
hipi:/tmp> cat /tmp/uname.log
HP-UX hipi A.09.05 A 9000/712 2000203734 two-user license
```

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

REMOTE_LOG_FILE

Syntax: `remote_log_file=`*`file`*

The `remote_log_file` qualifier specifies an output file to which the output of the `remote_command` qualifier should be redirected. The remote command is issued on the remote node when transfer is completed. The root directory for the file is the home directory of the user under which you logged in on the remote node.

The `remote_log_file` qualifier can only be used with the `remote_command` qualifier.

See also: `EXECUTE_ONLY` (227), `REMOTE_COMMAND` (255), `REMOTE_PARAM` (258) and `REMOTE_SYNC` (262).

Attributes

OVERRIDE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

REMOTE_PARAM

Syntax: `remote_param=`*`parameter_value`*

When the command specified by the `remote_command` qualifier is activated, the parameter_value specified by the `remote_param` qualifier is passed to the command on the remote node.

The parameter_value is a string of characters that can include variables and that are interpreted by FASTCopy prior to the activation of the remote command.

If the parameter_value contains spaces it must be enclosed in quotes.

The following variables are valid for the parameter_value:

%s Source filename of the last file that was transferred.

%t Target filename of the last file that was transferred.

%f Last file that was transferred; With remote_param, this is converted to the remote filename.

%1 Converts the succeeding filename to the format of the system where the remote_command is executed, rather than being translated on the local machine. This is a very useful modifier during cross-platform operations.

See also: `EXECUTE_ONLY` (227), `REMOTE_COMMAND` (255), `REMOTE_LOG_FILE` (257), `REMOTE_SYNC` (262) and `REMOTE_QUEUE` (261).

Note: From version 2.3.3 and above, this qualifier replaces the `p_remote_command` qualifier, which is now deprecated.

Examples

Assume that you are transferring all the files matching **a.*** from your current working directory, the contents of which are listed below:

UNIX

```
/usr/develop/yossi/fastcopy/src> ls -l a.*
-rw-r.r.   1 root   other            5 Jan 21 16:02 a.a
-rw-r.r.   1 root   other         1074 Sep 11 15:36 a.log
```

The following FASTCopy command will transfer both files and issue a remote command that will relate to the last file transferred. Specifying a variable in the `remote_param` makes it possible to issue the command without knowing at the beginning of the operation which file will be transferred last.

UNIX

```
/usr/develop/yossi/fastcopy/src> fcopy "a.*" hipi:/tmp -replace
-remote_command=\”ls,/tmp/ls.log\” -remote_param=\”-l %t\”
FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/fastcopy/src/a.a'
FASTCopied to file 'hipi:/tmp/a.a' ( 5 Bytes )
FCOPY-S-FILE_COPIED - File '/usr/develop/yossi/fastcopy/src/a.log'
FASTCopied to file 'hipi:/tmp/a.log' ( 1074 Bytes )
FCOPY-I-L_COMMAND - Remote command initiated
```

The output file of the remote command reveals that the variable in the `remote_param` was converted to the name of the last transferred file and contains the output of the *ls* command for that file.

UNIX

```
hipi:/tmp> cat /tmp/ls.log
-rw-r-r-   1 root     sys         22 Feb 20 15:32 /tmp/a.log
hipi:/tmp>
```

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

REMOTE_QUEUE

Syntax: `remote_queue=`*`queue`*

The `remote_queue` qualifier submits a FASTCopy remote command for execution under the named queue rather than the default OpenVMS batch queue.

This qualifier can be used only if the remote node is an OpenVMS platform, where FASTCopy's post-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `EXECUTE_ONLY` (227), `REMOTE_PARAM` (258) and `REMOTE_COMMAND` (255).

Attributes

BATCH
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	No
OpenVMS	Yes

REMOTE_SYNC

Syntax: `remote_sync`

The `remote_sync` qualifier specifies that FASTCopy should wait for a remote command to be completed before ending the operation; by default, FASTCopy considers a remote command successful if it was successfully issued. When `remote_sync` is used, the remote command is considered successful only if it returned with a status of 0. Otherwise, FASTCopy attempts recovery, issuing the command again.

This qualifier is not supported on OpenVMS platforms, where FASTCopy's post-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `EXECUTE_ONLY` (227), `REMOTE_COMMAND` (255) and `REMOTE_PARAM` (258).

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

REMOTE_PRE_COMMAND

Syntax: `remote_pre_command=`*`command`*

The `remote_pre_command` qualifier issues a command on the remote node, before starting a file transfer operation. The FASTCopy operation continues only after the the command is successfully completed. To enable the FASTCopy operation to continue before the local pre command is completed, use the negated form of the `REMOTE_PRE_SYNC` qualifier (`noremote_pre_sync`).

On UNIX, the command value can be a system command or an executable script. On OpenVMS, the command value must specify a command file (ending in **.com**). On Windows NT and Windows 9x/2000/XP, the command value must be the name of an executable file (ending in **.bat** or **.exe**).

You can specify an output file to which the command's output will be redirected using the `remote_pre_log_file` qualifier. The root directory for the file is the home directory of the user under which you logged in on the remote node.

Parameters can be passed to the command with the `remote_pre_param` qualifier.

The remote command is executed with the permissions of the user you specify when connecting to the remote node, but programs or scripts do not have access to any of that user's environment variables. These are set only when an actual login takes place.

See also: `EXECUTE_ONLY` (227), `REMOTE_PRE_PARAM` (265), `REMOTE_PRE_LOG_FILE` (264), `REMOTE_PRE_SYNC` (267) and `REMOTE_PRE_QUEUE` (266).

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

REMOTE_PRE_LOG_FILE

Syntax: `remote_pre_log_file=`*`file`*

The `remote_pre_log_file` qualifier specifies an output file to which the output of the remote pre-transfer command is redirected. The `remote_command` is issued on the remote node before transfer begins. The root directory for the file is the home directory of the user under which you logged in on the remote node.

The `remote_pre_log_file` qualifier can only be used with the `remote_pre_command` qualifier.

See also: EXECUTE_ONLY (227), REMOTE_PRE_COMMAND (263) REMOTE_PRE_PARAM (265) and REMOTE_PRE_SYNC (267).

Attributes

OVERRIDE
BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

REMOTE_PRE_PARAM

Syntax: `remote_pre_param=`*`parameter_value`*

When the `remote_pre_transfer` command is activated, the parameter_value specified by the `remote_pre_param` qualifier is passed to the command on the remote node.

The parameter_value is a string of characters that can include variables that are interpreted by FASTCopy prior to the activation of the `remote_pre_command`.

If the parameter_value contains spaces it must be enclosed in quotes.

See also: `EXECUTE_ONLY` (227), `REMOTE_PRE_COMMAND` (263), `REMOTE_PRE_LOG_FILE` (264), `REMOTE_PRE_SYNC` (267) and `REMOTE_PRE_QUEUE` (266).

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

REMOTE_PRE_QUEUE

Syntax: `remote_pre_queue=`*`queue`*

The `remote_pre_queue` qualifier submits a FASTCopy remote pre transfer command for execution under the named queue rather than the default OpenVMS batch queue.

This qualifier can only be used if the remote node is an OpenVMS platform, where FASTCopy's pre-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `EXECUTE_ONLY` (227), `REMOTE_PRE_PARAM` (265) and `REMOTE_PRE_COMMAND` (263).

Attributes

BATCH
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	No
OpenVMS	Yes

REMOTE_PRE_SYNC

Syntax: `remote_pre_sync`

The `remote_pre_sync` qualifier specifies that FASTCopy should wait for a remote pre transfer command to be completed before starting the transfer; by default, FASTCopy considers a remote pre transfer command successful if it was successfully issued. When `remote_pre_sync` is used, the remote pre transfer command is considered successful only if it returns with a status of 0. Otherwise, FASTCopy attempts recovery, issuing the command again.

This qualifier is not supported on OpenVMS platforms, where FASTCopy's pre-transfer processing commands are implemented under the OpenVMS batch queues.

See also: `EXECUTE_ONLY` (227), `REMOTE_PRE_COMMAND` (263) and `REMOTE_PRE_PARAM` (265).

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

RETRIEVE_REMOTE_LOG

Syntax: `retrieve_remote_log=`*`filename`*

The `retrieve_remote_log` qualifier transfers a copy of the remote log to the local machine on completion of the remote command. It should be used in conjunction with the `remote_sync` and `remote_log_file` qualifiers.

See also: `REMOTE_SYNC` (262) and `REMOTE_LOG_FILE` (257)

Attributes

OVERRIDE
NEGATABLE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

Connection Qualifiers

This section details qualifiers that let you specify commands that will be executed before and after each instance that FASTCopy connects to a remote node. A typical use of these qualifiers is to establish and disconnect a dial-up link. Because FASTCopy operations are handled differently on each supported platform, support for these qualifiers varies across platforms.

Connection qualifiers include:

- `LINK_DOWN_COMMAND` (270)
- `LINK_DOWN_PARAMETER` (271)
- `LINK_UP_COMMAND` (272)
- `LINK_UP_PARAMETER` (273)

LINK_DOWN_COMMAND

Syntax: `link_down_command=`*`command`*

The `link_down_command` qualifier specifies a command that will be carried out locally in the event that a FASTCopy batch operation fails. The command is executed after the first failure and after every unsuccessful recovery attempt that follows, and takes place before the operation enters a waiting phase. The `link_down_command` qualifier is intended for use in situations where a command must be executed locally after transfer takes place, such as when communication is through a dial-up connection. In these cases, the *command* specified with `link_down_command` is used to close the connection, established using the `link_up_command` qualifier, after each transfer attempt. If the operation terminates (either successfully or in unsuccessfully), the *command* specified by the `link_down_command` qualifier will not be carried out.

The `exit_command` qualifier must be used to disconnect the line.

Parameters can be passed to the command with the `link_down_parameter` qualifier.

See also: `INIT_COMMAND` (234), `EXIT_COMMAND` (229), `LINK_UP_COMMAND` (272) and `LINK_DOWN_PARAMETER` (271).

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

LINK_DOWN_PARAMETER

Syntax: `link_down_parameter=`*`parameter`*

The `link_down_parameter` qualifier transfers the specified parameter to a locally executed command initiated by the `link_down_command` qualifier before the FASTCopy batch operation enters a waiting phase.

See also: `LINK_DOWN_COMMAND` (270)

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

LINK_UP_COMMAND

Syntax: `link_up_command=`*`command`*

The `link_up_command` qualifier specifies a command that will be carried out locally before the beginning of a FASTCopy batch operation and before every recovery attempt made on that operation. The `link_up_command` qualifier is intended for use in situations where a command must be executed locally before transfer can take place, such as when communication is through a dial-up connection. In these cases, the `link_up_command` qualifier allows you to establish a communications line before each transfer attempt. The `link_down_command` qualifier can also be included to close the connection after each attempt.

Parameters can be passed to the command with the `link_up_parameter` qualifier.

See also: `INIT_COMMAND` (234), `EXIT_COMMAND` (229), `LINK_DOWN_COMMAND` (270) and `LINK_UP_PARAMETER` (273).

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

LINK_UP_PARAMETER

Syntax: `link_up_parameter=`*`parameter`*

The `link_up_parameter` qualifier transfers the specified parameter to a locally executed command specified with the `link_up_command` qualifier. The `link_up_command` qualifier is initiated before the beginning of a FASTCopy batch operation and before every recovery attempt made on that operation.

See also: `LINK_UP_COMMAND` (272)

Attributes

BATCH

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

Monitor Qualifiers

This section details qualifiers that affect the local and remote monitoring of FASTCopy operations. Because FASTCopy operations are handled differently on each supported platform, support for these qualifiers varies across platforms.

Monitor Qualifiers include:

- COMMENT (275)
- MONITOR (276)
- MON_LEVEL (277)
- MON_NODE (279)
- MON_TYPE (280)
- NEW_MONITOR (281)

COMMENT

Note: On Windows systems, this qualifier is supported from version 2.6 upwards.

Syntax: `comment=`*`text`*

The `comment` qualifier allows you to specify a string of text that will be sent by a monitored FASTCopy process to the monitoring node, along with other messages the process generates. A comment is specific to a given operation, allowing easier tracing and identification of the monitored operation on the monitoring node.

This qualifier is relevant only if the progress of this FASTCopy operation is reported to a central monitoring node. This is determined either by the settings specified in the **Security and Administration Files** or manually by specifying a central monitoring node with the `mon_node` qualifier.

See also: `MON_NODE` (279), `MON_LEVEL` (277) and `MON_TYPE` (280). In addition, refer to chapter 9 of *FASTCopy User's Guide*. Also refer to *FASTCopy Release Notes V2.6* for a description of the SoftLink Monitor GUI.

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

MONITOR

Syntax: `monitor`

The `monitor` qualifier instructs FASTCopy to report its log messages and progress to *flogicd*. The FASTCopy operation submits a batch job and sends the relevant messages to this job. The batch job number is shown in the FASTCopy messages. Recovery attempts of the FASTCopy operation continue to report to the same job.

The *flogicd* batch daemon provides centralized monitoring and control over the FASTCopy operation. The FASTCopy operation progress and current status are displayed in the job's status line.

Refer to the *FASTCopy Monitoring Guide* for more information on FASTCopy's monitoring capabilities. Refer also to chapter 4, *Security and Administration* in the FASTCopy *Administrator's Guide* for details on creating monitoring jobs for FASTCopy operations initiated by other users.

Attributes

POSITIONAL
OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

MON_LEVEL

Note: On Windows systems, this qualifier is supported from version 2.6 upwards.

Syntax: `mon_level=[min|default|detailed|verbose]`
Default: `mon_level=default`

The `mon_level` qualifier controls the level of detail that will be displayed on the SoftLink Monitor about the progress of the specified FASTCopy operation. If this qualifier is omitted, the default value of `default` is chosen.

This qualifier is significant only if the progress of the FASTCopy operation is reported to a central monitoring node. This is determined by performing one of the following:

- Entering the appropriate settings in the **Security and Administration Files**. (Refer to the *FASTCopy Administrator's Guide* for further information).
- Specifying a central monitoring node with the `mon_node` qualifier (UNIX).
- Configuring the **Send To** and **Receive From** tabs in the Monitor Configuration dialog box (Windows).

The monitoring level can be one of the following:

min Only the start and end of the FASTCopy operation is reported to the central monitor.

default The following stages of the FASTCopy operation will be reflected in the central monitor:

The start of the FASTCopy operation.

The end of the transfer stage.

Local command activation.

Remote command activation.

End of the FASTCopy operation.

detailed All the above events are reported to the SoftLink monitor, as well as the start and end information for each file that is transferred.

verbose Similar to **detailed**, but with the addition of packet and file compression data.

See also: `COMMENT` (275), `MON_NODE` (279) and `MON_TYPE` (280).

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

MON_NODE

Note: On Windows systems, this qualifier is supported from version 2.6 upwards.

Syntax: `mon_node=`*`node_name`*

-Or-

`mon_node=`*`“node_name, node_name, ...”`*

The `mon_node` qualifier is used to specify a central monitoring node (or nodes) that will be updated about the progress of the current FASTCopy operation.

This qualifier is significant only if you wish the progress of this FASTCopy operation to be reported to a central monitoring node. This can determined either by the settings specified in the **Security and Administration Files** or manually by specifying a central monitoring node with the qualifier `mon_node` in the command line. Command line monitoring directives may be overridden by the contents of the **Security and Administration Files**.

See also: `COMMENT` (275), `MON_LEVEL` (277) and `MON_TYPE` (280).

Attributes

OVERRIDE
LIST_VALUE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	No

MON_TYPE

Syntax: `mon_type=`*`label`*

The `mon_type` qualifier specifies an identifying label for all the messages that the current FASTCopy process sends to a central monitoring node. On the central monitoring node, this label is used to determine which particular program should handle these messages.

This qualifier is significant only if the progress of the current FASTCopy operation is reported to a central monitoring node. This is determined either by the settings specified in the **Security and Administration Files** or manually by specifying a central monitoring node with the `mon_node` qualifier in the command line.

If the `mon_type` qualifier is not specified either in the **Security and Administration Files** or in the command line, all information about the progress of a monitored FASTCopy operation is directed on the central monitoring node to the image which handles messages labeled **fcopy**. The `mon_type` qualifier allows you to use different images to monitor different operations.

See also: `COMMENT` (275), `MON_LEVEL` (277) and `MON_NODE` (279).

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	No
Windows NT	No
Windows 2000	No
Windows XP	No
UNIX	Yes
OpenVMS	No

NEW_MONITOR

Note: Supported from FASTCopy version 2.6 upwards.

Syntax: `new_monitor`

The `new_monitor` qualifier instructs FASTCopy to send monitoring messages to the Softlink Monitor. The Softlink Monitor provides graphical monitoring of FASTCopy operations on local and remote nodes.

Runtime messages and error reports are sent to the the issuing (i.e. local) node, unless otherwise configured. You can configure the SoftLink Monitor (with the Monitor Configuration tool) to report to and/or receive reports from one or more remote nodes running FASTCopy.

IMPORTANT: The FASTCopy command line must include the `new_monitor` qualifier to enable monitoring via the Softlink Monitoring Utility.

Attributes

OVERRIDE

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	No
OpenVMS	No

Support Qualifiers

The following section contains qualifiers pertinent to FASTCopy product support.

INFORMATION

Syntax: `information`

The information qualifier is a stand alone qualifier used to obtain details about the currently installed FASTCopy version. This qualifier must be used without any additional parameters.

Example

Windows

```
C:\Program Files\SoftLink\FASTCopy>fcopy -info

FASTCopy Version Information:
=============================
FASTCopy version is 2.6. Internal build version : 0.42
FASTCopy compression methods supported: 27
FASTCopy creation date : Dec 27 2001 12:12:47

C:\Program Files\SoftLink\FASTCopy>
```

Attributes

NO_PARAMETERS

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

FLICENSE

Syntax: `flicense=check|view`

The `flicense` qualifier allows you to verify the validity of and view the content of your current FASTCopy license.

Example

The following example shows the ouput of the `flicense=check` command.

```
C:\Program Files\SoftLink\FASTCopy>flicense check
LICENSE-S-VERIFIED, license successfully verified
```

Example

The following example shows the ouput of the `flicense=view` command.

```
C:\Program Files\SoftLink\FASTCopy>flicense view

License <C:\Program Files\SoftLink\FASTCopy\lic_fastcopy.sl>:
#================ BEGIN LICENSE ================
[HGSFT3-6F4G32-123SES]
-lid=sl.bob
-enable=encryption
#================= END LICENSE ================
```

Attributes

NO_PARAMETERS

Supported Operating Systems

Windows 9x	Yes
Windows NT	Yes
Windows 2000	Yes
Windows XP	Yes
UNIX	Yes
OpenVMS	Yes

A

Setting Time & Time-Frame Expressions

The qualifiers `before`, `since`, `hold_until` and `terminate_after` accept time expressions as values. These time expressions are given using the common syntax of the operating system. Time expressions are used to specify when a batch operation should begin or end, or to select files according to their creation or modification time.

Absolute Time Expressions

On all supported systems, you can specify a date or time in the following formats:

Time Format

Format	**Description**
hhmm	Hours (two digits) and minutes (two digits). The current date (**`today`**) is assumed.

Date Format

Format	**Description**
ddmmyy	Day of the month (two digits), month (two digits) and year (two digits). The time is assumed to be midnight on the specified date.

Time and Date Format

Format	Description
ddmmyyhhmm	A combination of the two expressions. You can also use the “today” keyword. If no time is specified with a qualifier that requires a time expression, the time is assumed to be midnight and the date is assumed to be today.

Additional OpenVMS Formats

On OpenVMS systems the following additional formats are also supported:

Time Format

Format	Description
HH:MM:SS.CC	Two digits each for hour, minutes, seconds and hundredths of a second. If omitted, the last two values (*SS* and *CC*) are both assumed to be 00. The current date (today) is assumed.

Date Format

Format	Description
DD-MMM-YYYY	Day of the month (two digits), month of the year (three letters), and year (four digits). The time is assumed to be midnight on the specified date.

Time and Date Format

Format	Description
DD-MMM-YYYY:HH:MM:SS.CC	A combination of the two expressions. You can also use the “today” keyword. If no time is specified, FASTCopy assumes that the date is today.

Relative Time Expressions

Sometimes it is more useful to specify a time expression as relative to the current time rather than in absolute terms. For example, if you want to select all files that are over a month old, or to specify that a particular operation will terminate three hours after it is initiated.

Relative time expressions are specified as values that will be added or subtracted from the current time to obtain the final value. They consist of one or more terms that are composed of a plus (+) or minus (-) sign, a numeric value, and a keyword indicating the unit.

The following units can be used:

- Years
- Months
- Weeks
- Days
- Minutes
- Hours
- Seconds

You must include a space between the value and the unit and always enclose the relative time expression in quotation marks.

If no sign is included in a term, it is considered to have a sign of +. Each term in an expression is added or subtracted (according to its sign) from the current time when the expression is evaluated (during execution).

Examples

Anything over six months old:

“-6 months”

Anything that is dated 3 weeks, 4 days and 7 hours before the current time:

“-3 weeks -4 days -7 hours”

Note that if a sign is not used specifically, the term has a sign of +, so that:

“-5 weeks 4 days”

Means 5 weeks less 4 days ago and not 5 weeks and 4 days ago. It is the same as the expression:

“-5 weeks +4 days”

or the expression:

“-4 weeks -3 days”

Setting Time Frames

Certain qualifiers can also accept **Time Frame Expressions** as values. These expressions specify when the transfer should be performed. A simple Time Frame Expression includes a range of times and a character that identifies their type, such as hours, months, or days of the week. You can use a "range" which includes only a single value, and you can omit either the upper or lower limit, specifying that the qualifier should take affect only *until* a certain time or *from* a certain time. You can also use a list of values instead of a range.

Valid time frame types include:

Character	**Type**
y	Years
M	Months (valid values range from 1 to 12)
d	Days of the month (valid values range from 1 to 31)
w	Days of the week (valid values range from 1 to 7)
h	Hours (valid values range from 1 to 24)
m	Minutes (valid values range from 1 to 60)
s	Seconds (valid values range from 1 to 60)
t	Time - uses 2, 4, 6 or 10 digit format and can replace h, m and s. ▪ 2 digit format = hours; ▪ 4 digit format = hours & minutes; ▪ 6 digit format = hours, minutes & seconds; ▪ 10 digit format = day of the month, month, hours, minutes, seconds.

Compound Time Frame Expressions

Simple Time Frame Expressions can be combined into compound expressions by stringing them together. An ampersand mark (&) between two expressions combines the expressions into a single time frame. If there is nothing between the two expressions, then one or the other may apply. Parentheses () can be used to enclose expressions, indicating that the expression in the parentheses is a single expression. An expression can also be defined as an exception - by adding an exclamation mark (!) before a given expression, you specify any time except that within the time frame that the expression describes.

If an hours range is specified with an upper limit that is smaller than the lower limit, for example, **`h:216`**, the upper limit is assumed to be on the next day. Thus, **`h:216`** is interpreted as a time frame starting at 9PM in the evening and ending at 6AM the following morning.

Examples of Compound Time Frame Expressions:

Time Period	**Compound Time Frame Expression**
Monday through Friday 8am-5pm.	(w:2-6)&(h:8-17)
Same as above, but excluding both new year's day and Christmas.	(w:2-6)&(h:8-17)&!((M:1)&(d:1) (M:12)&(d:25))
Every other day from 9pm onwards.	(w:2 4 6)&(t:2100-)
Every day before 8am and after 5pm.	(h:-8 17-)

B

Encryption Methods

FASTCopy allows you to encrypt file data that is transported over the network using the four qualifiers: `line_encrypt`, `line_phrase`, `line_cipher` and `cipher` (SSL). The `line_cipher` and `cipher` (SSL) qualifiers are used to specify which encryption method should be used, using a list of keywords.

The encryption methods use the following algorithms. Each algorithm uses one or more keys generated from the phrase specified by the user with the `line_phrase` or `cipher` qualifiers:

Algorithm	**Key Size**	**Description**
DES	56 bits	Data Encryption Standard
DES-EDE	2 x 56 bits (112 bits)	Encrypt-Decrypt-Encrypt, using DES and two different keys.
DES 3-EDE	3 x 56 bits (168 bits)	Encrypt-Decrypt-Encrypt, using DES and three different keys.
DESX	56 bits + 64 bits (120 bits)	“Whitening” is used to obscure input and output of DES
Blowfish	Variable-length	Fast and efficient
CAST	64 bits	The CAST Algorithm (Carlisle Adams, Stafford Travares)
RC2	Variable-length	Rivest Cypher encryption
RC4	Variable-length	Rivest Cypher encryption

DES based algorithms are used in combination with one of the following modes. The mode specifies how the given algorithm is used to encrypt that data. The possible modes are:

Mode	**Description**
ECB (Electronic Codebook)	Algorithm employed to encrypt discreet blocks.
CBC (Cipher Block Chaining)	Encryption of previous blocks affects the encryption of the current block.
CFB (Cipher Feedback)	As above, except that data is encrypted as a stream.
OFB (Output Feedback)	Data is encrypted as a stream; the algorithm's internal feedback affects further encryption.

The following table shows what algorithm and mode are used in each encryption method, and what keywords can specify that method. You can use different keywords to specify the method you want, but do not abbreviate any of these keywords:

		Mode			
		CBC	**ECB**	**CFB**	**OFB**
	DES	des, des-cbc	des-ecb	des-cfb	des-ofb
	DES-EDE	des-ede, des-ede-cbc		des-ede-cfb	des-ede-ofb
	DES-3-EDE	des3, des-ede3, des-ede3-cbc		des-ede3-cfb	des-ede3-ofb
	DESX	desx, desx-cbc			

If not instructed otherwise, FASTCopy will use DES-CBC as its encryption method. The merits and drawbacks of this and other encryption methods is beyond the scope of this documentation. Interested parties are invited to consult a cryptography textbook for further details.

Note: To implement this feature, you require a license that supports encryption.

The table below indicates the encryption modes supported by the Blowfish, CAST, RC2 and RC4 algorithms. The table items are acceptable values for the `line_cipher` or `cipher` qualifiers and must not be abbreviated.

		Mode			
		CBC	**CFB**	**ECB**	**OFB**
	Blowfish	bf-cbc	bf-cfb	bf-ecb	bf-ofb
	CAST	cast-cbc cast5-cbc	cast5-cfb	cast5-ecb	cast5-ofb
	RC2	rc2-40-cbc rc2-64-cbc rc2-cbc	rc2-cfb	rc2-ecb	rc2-ofb
	RC4	rc4-40			

See also: `LINE_CIPHER` (168), `LINE_ENCRYPT` (170), CIPHER (182) and LINE_PHRASE (172).

C

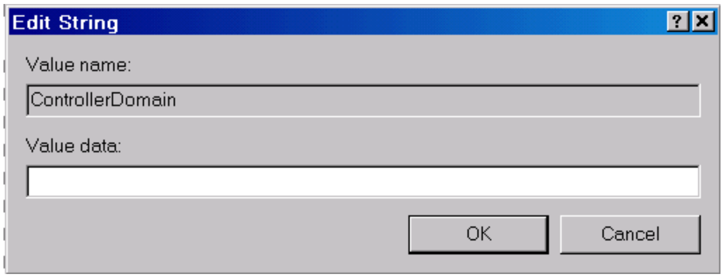
Saving Batch Qualifiers in the Registry

As of FASTCopy V2.6, you can save mandatory batch job qualifiers in a Windows registry file. Consequently, when you submit a batch job, Windows will automatically retrieve the qualifiers from the registry. This option is especially useful if you run FASTCopy batch jobs on a regular basis, since it allows you to exclude the `controller_user`, `controller_password` and `controller_domain` qualifiers from the command line.

Creating a Batch Job Registry File

To create a Batch Job Registry File:

1. Open the Windows Registry Editor.
2. Under HKEY_CURRENT_USER, right-click the **Software** key and select **New** and then **Key** from the pop-up menu.



Value name: ControllerDomain

Value data:

3. Rename the new key "Softlink".
4. Repeat steps 1-3 until the following path is created:

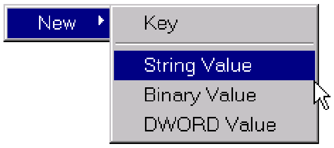
HKEY_CURRENT_USER/Software/Softlink/FastCopy/CurrentVersion.

Important: Key names must appear exactly as specified in the above path.

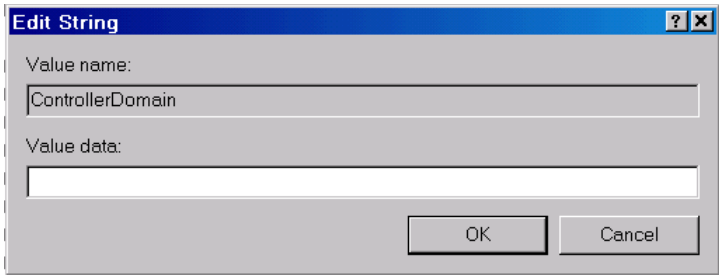
Note: If you previously installed FASTCopy V2.5, the **Softlink** and **FastCopy** keys may already exist.

To add the Batch qualifiers to the Current Version key:

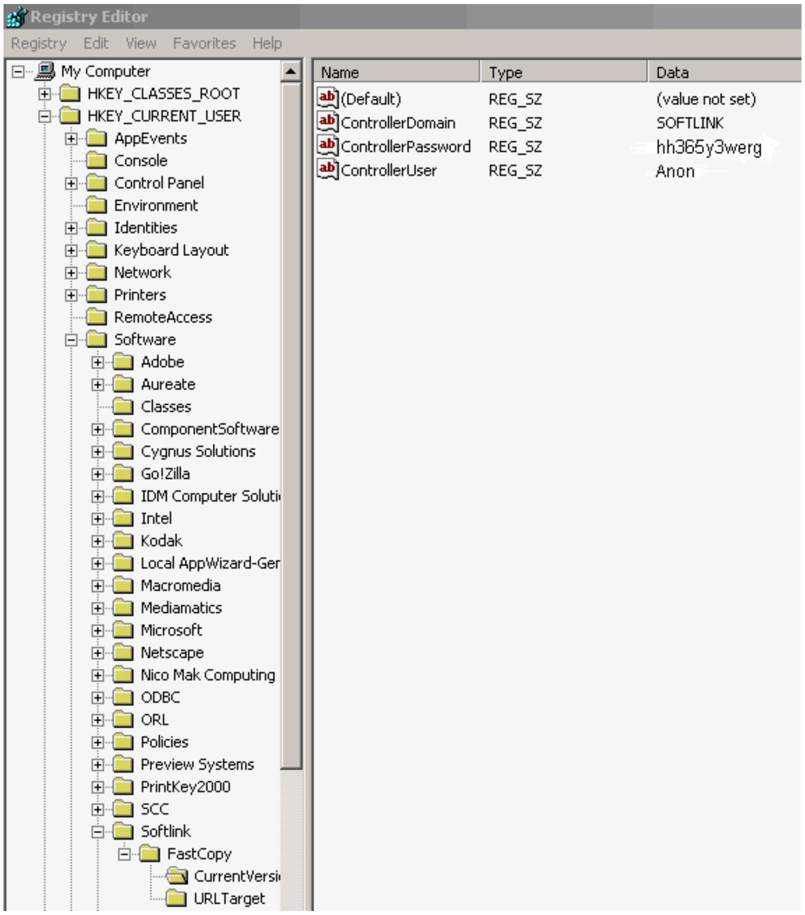
1. Right-click the right pane of the Registry Editor and select **String Value.**



2. Rename the **String Value** "ControllerDomain".
3. Double-click **ControllerDomain** and enter the local domain name in the **Value data** field. If the Controller User is a local system account, enter "NODOMAIN" in this field.



4. Repeat steps 1 to 3 to add the **ControllerPassword** and **ControllerUser** String values and data.
   a. Enter a scrambled password for the ControllerPassword data. (For instructions on generating a Scrambled Password, see the GENERATE (34) qualifier).
   b. Enter your username for the ControllerUser data (see figure below).



```
My Computer
  HKEY_CLASSES_ROOT
  HKEY_CURRENT_USER
    AppEvents
    Console
    Control Panel
    Environment
    Identities
    Keyboard Layout
    Network
    Printers
    RemoteAccess
    Software
      Adobe
      Aureate
      Classes
      ComponentSoftware
      Cygnus Solutions
      Go!Zilla
      IDM Computer Soluti
      Intel
      Kodak
      Local AppWizard-Ger
      Macromedia
      Mediamatics
      Microsoft
      Netscape
      Nico Mak Computing
      ODBC
      ORL
      Policies
      Preview Systems
      PrintKey2000
      SCC
      Softlink
        FastCopy
          CurrentVersi
          URLTarget
```

Name	Type	Data
(Default)	REG_SZ	(value not set)
ControllerDomain	REG_SZ	SOFTLINK
ControllerPassword	REG_SZ	hh365y3werg
ControllerUser	REG_SZ	Anon

Figure 1 – FASTCopy Batch job qualifiers in the Current Version registry key

After you have added the values to the registry you can submit a FASTCopy batch job without including the `controller_user`, `controller_password` and `controller_domain` qualifiers.

Batch Job Hierarchy

Command line qualifiers take precedence over qualifiers listed in the registry file. In other words, a `controller_user` qualifier included in the command line will override the **ControllerUser** value listed in the registry file. Any mandatory batch mode qualifier that is not included in the registry file must be specified in the FASTCopy batch job command line.

For instance, if you only specified the **ControllerUser** in the registry file, you would need to specify `controller_domain` and `controller_password` (or `scrambled_controller_password`) in the command-line.

Including a Scrambled Password in the Batch Job Command Line

To use a scrambled password in a batch job:

- Replace the `controller_password` qualifier with the `scrambled_controller_password` qualifier.

See also: `CONTROLLER_DOMAIN` (206), `CONTROLLER_PASSWORD` (207), `CONTROLLER_USER` (208) and `GENERATE` (34).

D

FASTCopy Command Line QuickStart

In simple terms, FASTCopy transfers files between two machines (nodes) residing on either a LAN or a WAN. The node from which the file(s) originates is called the source node, and the node to which the files are copied is called the destination or target node. One of these two nodes must be the local node, and the other must be a remote node. The FASTCopy application *must be installed on both the local and the remote nodes.*

FASTCopy can either transfer files from the local node to a remote node (push) or from a remote node to the local node (pull). Using a command line, FASTCopy cannot transfer files between two remote nodes, though this is possible using FASTCopy Console.

This chapter illustrates how to perform FASTCopy file transfers and is divided into the following sections:

- Invoking FASTCopy
- Same platform file transfers
- Cross-platform file transfers

Invoking FASTCopy Commands

FASTCopy command line operations are invoked through a UNIX shell, VMS DCL, or Windows MS-DOS command line interface. A FASTCopy command is invoked by typing the command **`fcopy`** followed by two *mandatory* parameters, two mandatory qualifiers and any number of *optional* qualifiers.

IMPORTANT: FASTCopy commands must be issued from the FASTCopy working directory. Always remember to change the path to the FASTCopy working directory.

The default Windows installation directory is:

C:\ProgramFiles\SoftLink\FASTCopy\bin>

Mandatory Parameters

The mandatory parameters are as follows:

1. The source file specification(s)
2. The destination file specification

One of these two parameters must also include the name or IP address of the remote node. If the remote node is the source node, it is included with the source file specification(s); if the remote node is the target node, it is included with the destination file specification.

For example, to transfer the file **car.exe** from the **c:\temp** directory of the local node to the **c:\temp** directory of the **nt13** remote node (assuming that both nodes are running Windows), issue the following command:

```
>fcopy c:\temp\car.exe \\nt13\c:\temp\car.exe –user=sam -pass=sam
```

Note: FASTCopy can also transfer files between directories on the local machine. Although not commonly used for this purpose, it may be helpful to practice issuing FASTCopy commands locally.

Optional Qualifiers

The two parameters may be followed by any number of optional qualifiers. Qualifiers serve to activate and control FASTCopy's numerous special features. These range from simple options, such as various ways to select files, to sophisticated capabilities, such as automatic file transfer recovery, guaranteed transfer scheduling and post-transfer processing. Understanding the functionality and limitations of each qualifier is necessary if you want to utilize the full potential of FASTCopy. For a detailed description of qualifiers, see page 306, Using FASTCopy Qualifiers.

Mandatory Qualifiers

In addition to the source and destination file specifications, every FASTCopy file transfer command *must* include a **username** and **password**. The username and password are required to access the remote node or, in the case of a local operation, the local node. The username and password may either be provided by the user in the FASTCopy command line, or defined by the system administrator in the FASTCopy **Security and Administration Files** residing on the remote/local node.

Username and password qualifiers

Scenario	Qualifiers
Password and username are *not* specified in the **Security and Administration Files**.	`user=`*`username`* `pass=`*`password`*
Password and username *are* specified in the **Security and Administration Files**.	`nouser nopass`

Although, usually appearing at the end of the command line, the username and password qualifiers can appear anywhere after the **`fcopy`** command (except in the middle of a file specification).

IMPORTANT: A FASTCopy file transfer operation *will not work* without a valid username and password. However, for simplicity's sake, most of the examples in this section do not include the username and password qualifiers. These must be added should you wish to test the example commands.

Refer to the *FASTCopy Administrator's Guide* for a comprehensive description of FASTCopy security files.

FASTCopy Command Line Syntax

The following general FASTCopy command syntax applies to all supported systems.

To copy files from a local node to a remote node:

```
> fcopy source_file_spec dest_node:dest_file_spec [qualifiers]
```

Alternatively, to copy files from a remote node to a local node:

```
> fcopy source_node:source_file_spec dest_file_spec [qualifiers]
```

The table below describes the FASTCopy command parameters.

Parameter	Description
source_file_spec	The file(s) that you wish to transfer. The source file specification can comprise of multiple file specifications, each with its own positional qualifier When specifying multiple files, you must separate each individual file specification with a comma. If a file specification includes a positional qualifier, the comma must follow the positional qualifier. (See page 306, Using FASTCopy Qualifiers, for more information).
dest_file_spec	The destination of the files being transferred. If you omit the destination file specification, the source file(s) will be copied to the user's home directory on the destination node. If the source file specification matches that of a file or files already existing on the destination node, FASTCopy will prompt the user for overwrite permission. Note that in UNIX, If you use wildcards in the source file specification, you must include wildcard characters in the destination file specification that match the wildcards in the source file specification. (See page 314, Taming Wildcards, for more information.)
source_node	The node from which files are transferred.
dest_node	The node to which files are transferred.

qualifiers	Mandatory (see page 300, Mandatory Qualifiers) and optional command line qualifiers that modify FASTCopy operations.

You must specify a single remote node in the command line. This node can be either the source or destination node. You can either *transfer* files to, or *receive* files from a remote node.

Note: You cannot copy files between two remote nodes.

Using FASTCopy with No Parameters

In a few special cases, the **`fcopy`** command is issued without including any file or node specifications. These cases are as follows:

Case 1: Displaying FASTCopy version information.

```
>fcopy -info
```

Case 2: Manual recovery of a previous FASTCopy operation.

```
>fcopy –recover
```

Case 3: Retrieving information about a previous FASTCopy operation.

```
>fcopy –dump
```

Case 4: Executing a command on either the local or remote node.

```
>fcopy –execute_only -remote_command|-local_command
```

File Specifications Format

FASTCopy uses the syntax of the local operating system for both source and destination file specifications, even if – as is the case with cross-platform operations - the file specification syntax on the remote node is different. This means that when working on a UNIX machine, you specify files using UNIX syntax, on Open VMS machines, you must use Open VMS syntax, and on Windows machines, you must use Windows file specification syntax.

Below is a list of Supported Operating Systems and their different syntax conventions.

OPENVMS

Syntax	`node::disk:[dir1.dir2.....]name.type;version`
Example	`zoltar::disk0[tmp]a.txt;3`

WINDOWS

Syntax	`\\node\disk:\dir1\dir2\.....\name`
Example 1	`\\nt13\c:\tmp\a.txt`
Example 2	`\\123.123.123.88\c:\tmp\a.txt`

UNIX

Syntax	`node:/dir1/dir2/…../name`
Example	`sun:/tmp/a.txt`

Note: Unlike Windows or OpenVMS syntax conventions, device or disk drives are not normally specified in UNIX file specifications. When working on a UNIX node opposite one of the other platforms, the drive must be specified with the term `dev/` preceding it in the path.

In the folloeing example, files are being transferred from a UNIX to a Windows platform:

```
> fcopy /tmp/a.txt nt13:/dev/c/tmp/a.txt
```

Cross-Platform Operations

In all of the examples below, the file **a.txt** in the directory **tmp** (which resides in the FASTCopy working directory) is being transferred from the local node to a remote node running a different operating system. The local file specification convention is used for both source and destination specifications. The node, directory and filenames are the same as in the previous examples.

UNIX to OpenVMS

Action:	Transfer from the UNIX node **sun** to the OpenVMS node **zoltar**.
Command:	`> fcopy /tmp/a.txt zoltar:/dev/disk0/tmp/a.txt`

UNIX to Windows

Action:	Transfer from the UNIX node **sun** to the Windows NT node **nt13**.
Command:	`> fcopy /tmp/a.txt nt13:/dev/c/tmp/a.txt`

OpenVMS to UNIX

Action:	Transfer from the OpenVMS node **zoltar** to the UNIX node **sun**.
Command:	`$ fcopy [tmp]a.txt;3 sun::[tmp]a.txt`

OpenVMS to Windows

Action:	Transfer from the OpenVMS node **zoltar** to the Windows NT node **nt13**.
Command:	`$ fcopy [tmp]a.txt;3 nt13::c:[tmp]a.txt`

Windows to OpenVMS

Action:	Transfer from the Windows NT node **nt13** to the OpenVMS node **zoltar**.
Command:	`c:> fcopy tmp\a.txt \\zoltar\disk0:\tmp\a.txt`

WINDOWS TO UNIX

Action:	Transfer from the Windows NT node **nt13** to the UNIX node **sun**.
Command:	`c:> fcopy tmp\a.txt \\sun\tmp\a.txt`

Note: When transferring files between an OpenVMS node and a UNIX or Windows node, remember that, unlike VMS, UNIX and Windows platforms are case-sensitive.

Using FASTCopy Qualifiers

A FASTCopy operation can be modified using optional qualifiers; these qualifiers are used to activate all of FASTCopy's special features, such as compression, verification, post-transfer processing, and delayed scheduling.

Qualifiers can either affect an entire operation, or individual source files. There are two types of qualifiers:

- Global Qualifiers
- Positional Qualifiers

Global qualifiers are typed at the end of the command line and affect the transfer of all files specified.
Positional qualifiers only affect individual source files and are specified in the source file specifications immediately after the file(s) that they affect.

Each qualifier must be preceded by a dash (-) on UNIX and Windows systems, or a slash (/) on OpenVMS systems. The dash/slash is immediately followed by the qualifier's name.

Abbreviating Qualifiers

You can use any abbreviated form of the qualifier name, provided that the abbreviation is unambiguous.

For example, the qualifier -**`report`** can be specified by -**`report`**, -**`repor`**, or -**`repo`**. However, you cannot use the abbreviation -**`rep`**, since FASTCopy will not be able to distinguish whether you meant -**`report`** or -**`replace`**.

A qualifier can either be specified on its own (-**`report`** and -**`replace`** are examples of such qualifiers), or with a value. The value is separated from the qualifier by an "equals" symbol (=). If the value is a string, it can also be abbreviated, again providing that the abbreviation is unambiguous.

For example:

```
-compression=normal
```

can also be written as:

```
-compr=norm
```

but not as:

```
-compr=no
```

since FASTCopy will not know whether you meant **`normal`** or **`none`**, both being valid compression values.

Positional and Global Qualifiers

Qualifiers that appear after the destination file specifications are global qualifiers and affect the entire FASTCopy operation. Qualifiers that are specified immediately following the source file specification are considered positional and affect FASTCopy operations on the preceding file specification only. A positional qualifier can also be specified as global, in the same command but with a different value. When the same qualifier is both positional and global, the positional qualifier's value takes precedence over the global qualifier's value for that file specification. However, the global qualifier affects all other file specifications. For more information about positional and global qualifiers, see the examples later in this section.

Examples of Qualifier Usage

FASTCopy qualifiers can be used in a variety of different ways, examples of which are provided below.

- Specifying Multiple Source Files with Positional Qualifiers Only.
- Specifying Multiple Source Files with Positional and Global Qualifiers.
- Pulling a File from a Remote Node using Global Qualifiers.

Some of the following examples include wildcards. If you are not familiar with wildcard usage, see Taming Wildcards (314).

Specifying Multiple Source Files with Positional Qualifiers Only

UNIX

```
> fcopy "/usr/john/*.c -comp=power -veri=sync , /usr/bob/*.h -
comp=none" "remote:*.back"
```

Windows

```
c:> fcopy \usr\john\*.c -comp=power -veri=sync , \usr\bob\*.h -
comp=none \\remote\*.back
```

OpenVMS

```
% fcopy [usr.john]*.c /comp=power /veri=sync , [usr.bob]*.h /comp=none
remote::*.back
```

The above command performs the following twofold operation:

1. Copies all files matching **`*.c`** from the subdirectory **john** to the remote destination, using power compression and ensuring synchronized write of the output files for verification purposes.
2. Copies all files that match **`*.h`** from the subdirectory **bob** to the same destination, with no compression, and no verification.

Specifying Multiple Source Files with Positional and Global Qualifiers

UNIX

```
> fcopy "/usr/john/*.c -compression=power ,/usr/bob/*.h "
"remote:*.back" -comp=non -verify=compare
```

Windows

```
c:> fcopy \usr\john\*.c –compression=power ,\usr\bob\*.h
\\remote\*.back –comp=non –verify=compare
```

OpenVMS

```
% fcopy [usr.john]*.c /compression=power ,[usr.bob]*.h remote::*.back
/comp=non /verify=compare
```

This command performs the following threefold operation:

1. Copies all files matching ***.c** from the subdirectory **john** to the remote destination using **`power`** compression.
2. Copies all files matching ***.h** from the subdirectory **bob** to the same destination, with no compression.
3. Compares all files (from both source file specifications) after the transfer, since **`–verify=compare`** is positioned as a global qualifier.

Pulling a File from a Remote Node using Global Qualifiers

UNIX

```
> fcopy BOSTON:/usr/sam/project/test.dat my_test.dat -
compression=power -user=sam
```

WINDOWS

```
c:> fcopy \\BOSTON\usr\sam\project\test.dat my_test.dat –
compression=power –user=sam
```

OPENVMS

```
$ fcopy BOSTON::[usr.sam.project]test.dat my_test.dat
/compression=power /user=sam
```

The outcome of this command is as follows:

The file **test.dat** is copied from the project subdirectory on the **BOSTON** node to a new filenamed **my_test.dat** on the local node.

The -**`compression`** qualifier indicates that the **`power`** compression method will be used. The **`-user`** qualifier indicates that the login on the remote node will use the username **`sam`**. Since no password is specified, FASTCopy will prompt the user for a password.

Qualifiers with Multiple Values

Some qualifiers can accept more than one value. For example, the qualifier -**verify** can accept the values **sync** and **compare**. Multiple values must be enclosed in quotation marks (" ") and each quotation mark must be preceded either by a backslash (UNIX and Windows) or two more quotation marks (OpenVMS).

Examples

UNIX and Windows

```
-verify=\"sync, compare\"
```

OpenVMS

```
/verify="""sync, compare"""
```

General Examples

The following examples include wildcards. If you are not familiar with wildcard usage, see Taming Wildcards (314).

Transferring and Renaming a Single File

UNIX

```
> fcopy test.dat VENUS:newtest.dat
```

Windows

```
c:> fcopy test.dat \\VENUS\newtest.dat
```

OpenVMS

```
$ fcopy test.dat VENUS::newtest.dat
```

This command performs the following twofold operation:

1. Copies the **test.dat** file from the current working directory to the remote user's default directory on the **VENUS** node.
2. Renames it **newtest.dat**.

Transferring Multiple Files

UNIX

```
> fcopy "*" PRTLND:/tmp
```

Windows

```
c:> fcopy *.* \\PRTLND\tmp
```

OpenVMS

```
$ fcopy *.* PRTLND::[tmp]
```

This command copies all files within the user directory matching the wildcard **"*"** from the local node to the relative directory **tmp** on the **PRTLND** remote node.

The new files will have the same names as the copied files. Note the following:

- If you are executing the command from the UNIX C-Shell, you must enclose the file specifications in quotes to prevent the C-Shell from misinterpreting any wildcards.
- If you initiate FASTCopy through the Application Programming Interface from a user program, you *must* omit the quotes.

Transferring and Renaming Multiple Files

UNIX

```
> fcopy "*.c" "PRTLND:*.back"
```

OpenVMS

```
$ fcopy *.c PRTLND::*.back
```

Windows

```
c:> fcopy *.c \\PRTLND\*.back
```

This command performs the following twofold operation:

1. Copies all files matching the specification **`*.c`** in the local node's current working directory to the remote node **PRTLND**.
2. Renames all copied files ***.back**.

Specifying Multiple Source Files

UNIX

```
> fcopy "/usr/john/*.c,/usr/bob/*.h" "remote:*.back"
```

Windows

```
c:> fcopy \usr\john\*.c,\usr\bob\*.h" \\remote\*.back
```

OpenVMS

```
$ fcopy [usr.john]*.c,[usr.bob]*.h remote::*.back
```

This command performs the following threefold operation:

1. Copies all files matching ***.c** from the subdirectory **john** to the remote destination.
2. Copies all files matching ***.h** from the subdirectory **bob** to the same destination.
3. Renames all copied files ***.back**.

Taming Wildcards

FASTCopy supports the use of wildcards (keyboard meta-characters) in file specifications. This allows you to designate multiple files using a single meta-character. When selecting which files to transfer, you can include wildcards anywhere in the source file specification to select all files whose names match the given pattern.

To copy all files specified in the source file specifications to the destination and keep the same filenames:

Operating System	**Method**
OpenVMS or Windows	Type **`*`** or **`*.*`** in the destination file specification.
UNIX	Type **`*`** in the destination file specification.
All systems	Do not specify a filename in the destination file specification.

UNIX and Windows NT wildcards include **`?`** (any single character) and **`*`** (any string of characters).

Using Wildcards with UNIX Shells

On UNIX systems, if you use wildcards in a source file specification and include a destination file specification, the destination file specification must contain an equal number of wildcards as the source file specification. During transfer, when the target filename is parsed, the strings matching the wildcards in the source file specification replace the corresponding wildcards in the target filename.

When you use wildcards with the **`fcopy`** command in a UNIX shell environment, you must enclose the entire file specification (input and/or output) in double quotes or precede the wildcard character with a '\' (backslash). This prevents the shell from interpreting the command incorrectly.

Transfers Examples

This example assumes your source directory contains the following files: **aa, ab, a.a, abc.a, audrey, bb, bc.**

File Specification	**Matching Files**
a*	aa, ab, a.a, abc.a, audrey
a?	aa, ab

The table below displays the result of transferring **a*** to **new***.

Original Filename	Filename after Transfer
aa	newa
ab	newb
a.a	new.a
abc.a	newbc.a
audrey	newudrey

Using Wildcards with OpenVMS

OpenVMS wildcards include the wildcard **%** (any single character) and ***** (any string of characters within a field).

On OpenVMS systems, the target file specification can contain an ***** (asterisk) wildcard in either the name *or* the type field. This is replaced by the complete matching field in the source.

Note: Unlike UNIX, you cannot use strings containing wildcards in a field. You must use either a string *or* a wildcard.

Example Transfers

This example assumes your source directory contains the following files: **aa, ab, a.a, abc.a, audrey, bb, bc.**

File Specification	Matching Files
a*	aa, ab and audrey.
a*.*	aa, ab, audrey, a.a and abc.a.
a%	aa and ab.

The table below displays the result of transferring **a*.*** to **new.***.

Original Filename	Filename After Transfer
aa	new
ab	new
a.a	new.a

abc.a	new.a
audrey	new

Case Sensitivity in Filenames and File Specifications

Unlike UNIX and Windows, OpenVMS file naming is not case sensitive. When copying files from OpenVMS to UNIX or Windows NT, FASTCopy assumes that the target file will be created with a lower case or upper case name according to how the source specification was specified in the command. When copying from other systems to OpenVMS, filenames on OpenVMS will always be in upper case.

OpenVMS Examples

The file **A.A** resides in the current working directory.

If the following command is used:

```
$ fcopy a.* sun::[tmp]
```

FASTCopy will copy **A.A** from the local OpenVMS node to **/tmp/a.a** on the UNIX node **sun**.

If the user specifies:

```
$ fcopy A.* sun::[tmp]
```

The file **A.A** will be copied to **/tmp/A.A** on the UNIX node **sun**.

Windows filenames are case sensitive but the Windows file search is not case sensitive.

WINDOWS EXAMPLES

The files **A.A**, **B.a** and **c.a** are located in the current working directory.

If the following command is issued:

```
c:> fcopy *.a \\sun\tmp
```

FASTCopy will copy **A.A** to **/tmp/A.A**, **B.a** to **/tmp/B.a** and **c.a** to **/tmp/c.a** on the UNIX node **sun**.

If an output file specification such as the following is used:

```
c:> fcopy *.a \\sun\tmp\*.OLD
```

FASTCopy will notice that the source specification is in lower case and assume that upper case files were selected because they met the parameters of the Windows NT search but the intention was to use lower case.

It will copy **A.A** to **/tmp/a.OLD**, **B.a** to **/tmp/b.OLD** and **c.a** to **/tmp/c.OLD** on the UNIX node **sun**.

However, if the following specification is given:

```
c:> fcopy *.A \\sun\tmp\*.OLD
```

FASTCopy will notice that the source specification is in upper case and assume that the intention was to preserve the upper and lower case in the UNIX target filenames.

It will copy **A.A** to **/tmp/A.OLD**, **B.a** to **/tmp/B.OLD** and **c.a** to **/tmp/c.OLD** on the UNIX node **sun**.

E

Starting FASTCopy Daemon with the Required SSL Certificate and Key

The following section details the procedure for storing both parts (see below) of the encrypted private keyphrase in the Windows registry of the FASTCopy daemon machine.

Let’s assume that the following is true:

1. FASTCopy daemon private key resides in:

   c:\program files\softlink\ssl\fc_server_key.pem

2. The key phrase is:

   12345678

3. FASTCopy daemon certificate resides in:

   c:\program files\softlink\ssl\fc_server_cert.pem

Issue the following command:

```
fcopy –scramble_key_phrase –key=\”c:\\program
files\\softlink\\ssl\\fc_server_key.pem\” -key_phrase=12
345678 –export=c:\server.cf
```

The above command will cause FASTCopy to generate 2 scrambled tokens. One is called the scrambled key phrase and the other is the scrambled key material. They will be written to the file named **c:\server.cf** (the file name is a matter of choice of course).

4. Edit the file and make sure that in the line containing the key name, only the file name itself without the path is defined. For example:

   Key Name=fc_server_key.pem

5. Stop the fastcopy daemon service and add the following registry entry:

   a. Key:

      KEY_LOCAL_MACHINE\SOFTWARE\SoftLink\FASTCopy\CurrentVersion

   b. String value name:

      DaemonKeyList

   c. Value:

      c:\server.cf

6. Restart the daemon service.
7. Remove the previously defined registry entry: **DaemonKeyList**.
8. Add the following registry entry:

   a. Key:

      HKEY_LOCAL_MACHINE\SOFTWARE\SoftLink\FASTCopy\CurrentVersion

   b. String value name:

      PrivateKeyFile

   c. Value:

      c:\program files\softlink\ssl\fc_server_key.pem

Let’s assume that the following is true:

1. The client key file resides in:

   C:\program files\softlink\ssl\fc_client_key.pem

2. Client key phrase is

   12345678

3. Client certificate resides in:

   C:\program files\softlink\ssl\fc_client_cert.pem

Issue the following command:

```
fcopy a.a \\localhost\b.b -report -replace -user=someuser
-pass=somepass  -auth=cert –key=\”c:\\program
files\\softlink\\ssl\\fc_client_key.pem\”
-key_phrase=12345678 –cert=\”c:\\program
files\\softlink\\ssl\\fc_client_cert.pem\”
```

INDEX