
Robocopy.exe

Robust File Copy Utility

Version XP010

Robocopy is a 32-bit command-line tool used for file replication. This tool helps maintain identical copies of a directory structure on a single computer or in separate network locations. Robocopy is included in the *Microsoft® Windows® Resource Kit*.

Important Robocopy version XP010 requires Microsoft® Windows® Server 2003, Microsoft® Windows® XP, Microsoft® Windows® 2000, or Microsoft® Windows NT® version 4.0.

Using Robocopy, you can copy a single directory, or you can recursively copy a directory and its subdirectories. The tool classifies files by whether they exist in the source directory, in the destination directory, or in both. In the latter case, the tool further classifies files by comparing time stamps and file sizes between the source file and the corresponding destination file. You control which classes of files are copied.

If a file exists in both the source and destination locations, by default Robocopy copies the file only if the two versions have different time stamps or different sizes. This saves time if the source and destination are connected by a slow network link. You can also specify that copies are restarted in the event of a failure, which saves even more time when your network links are unreliable.

Robocopy allows you to do the following:

- Use file names, wildcard characters, paths, or file attributes to include or exclude source files as candidates for copying.
- Exclude directories by name or by path.
- Delete source files and directories after copying (that is, move rather than copy them).
- Delete destination files and directories that no longer exist in the source.
- Control the number of times the program retries an operation after encountering a recoverable network error.
- Schedule copy jobs to run automatically.
- Specify when copying is to be performed.
- Monitor a directory tree for changes.

- Selectively copy file data.

Contents

New Features in Robocopy Version XP010	3
Command-line Reference	7
Usage Notes	11
Backwards Compatibility	11
Using Robocopy Within a UNIX Shell	11
Processing a Directory Tree	11
Retrying After a Failed Copy	12
Specifying File and Directory Names	13
Using Robocopy File Classes	14
Using Precise File Times and Long File Names	15
Working with File Attributes	16
Maintaining True Replication	17
Selectively Copying File Data	18
Copying NTFS Security Information	18
Moving Files	19
Minimizing Directory Fragmentation	20
Robocopy Job Files	20
Scheduling Robocopy Jobs	23
Monitoring a Directory Tree	24
Controlling Bandwidth Usage	25
Output Log	26
Text Tags	27
Run Summary	30
Return Code	31
Feedback	32

New Features in Robocopy Version XP010

Robocopy version XP010 includes a number of enhancements. For example, you can now do the following:

- Copy Junctions – Volume Mount Points and Directory Links. By default these are copied as normal directories, as it may not be possible to create an equivalent Junction in the destination, especially if the source and destination reside on different machines. Copying a tree that contains a Volume Mount Point can result in unexpectedly large volumes of data being copied.

Use **/XJ** if you would prefer Robocopy to skip Junctions.

- Act on and modify more file attributes.

In addition to support for the R, A, S, and H attributes:

/XA and **/IA** now also accept and select on the following additional file attributes:

C – Compressed files
N – files Not content indexed
E – Encrypted files
T – Temporary files
O – Offline files

/A+ and **/A-** now also accept and modify the following additional attribute type:

N – files Not content indexed
T – Temporary files

- Select files based on their Last Accessed Dates.

/MAXLAD:n excludes files with a Last Access Date older than n.

/MINLAD:n excludes files with a Last Access Date newer than n.

If $n < 1900$, n is taken to be a time span in days.

If $n \geq 1900$, n must be a date in the form YYYYMMDD

- Copy a new class of file : “Tweaked” files.

Use **/IT** to include and process Tweaked files.

A Tweaked file is defined to be one that exists in both the source and destination,

with identical size and timestamp, but different attribute settings.

For more information see “Robocopy File Classes” later in this document.

- Copy very long path names (longer than 256 characters).
Previous versions of Robocopy would fail to copy any pathname longer than 256 characters. The program has now been enhanced to handle pathnames of any length, up to the internal Windows limit of nearly 32,000 characters.
- Get better information on data volumes for large copies.
Previous versions of Robocopy used 32-bit counters which would overflow when more than 2 gigabytes of data was processed, resulting in incorrect output in the run summary. This version now uses 64-bit counters to avoid this problem.
- More easily correlate Robocopy errors to network errors.
Robocopy now displays the date and time that errors occur to assist in tracking down communication problems.
- Recover more quickly and robustly from file copy failures.
This version of Robocopy now timestamps destination files with a January 1980 date during copies. Thus, if a copy of a newer file is interrupted and re-run, the destination file will seem to be older than the source file and be copied correctly.
- Copy files in Backup Mode.
Use the **/B** switch to copy all files in Backup Mode. Use the **/ZB** switch to try to copy files in Restartable Mode (for greater resiliency), and automatically switch to Backup Mode if the Restartable copy fails with an “Access Denied” error.

Note You must have the Backup Files and Restore Files privileges to copy files in Backup Mode using **/B** or **/ZB**.

Note Backup mode copies are not restartable, but they enable you to copy some files as a Backup Operator that you would not be able to copy as a normal user.

- Create, Edit, View, and take parameters from Robocopy Job (.RCJ) Files.
Use the **/SAVE:job** switch to save parameters to a Job file.

Use the **/JOB:job** switch to take parameters from one or more Job files.

Use the **/JOB:job** and **/SAVE:job** switches together to edit a Job file.

Use the **/QUIT** switch to end execution after processing command line arguments.

Use the **/JOB:job** and **/QUIT** switches together to view the contents of a job file.

Use the **/NOSD** to create a Job template with no Source Directory.

Use the **/NODD** to create a Job template with no Destination Directory.

Robocopy Job (.RCJ) files are simple text files that can also be edited manually.

For more information, see “Job Files” later in this document.

- Monitor the source directory tree for changes, and copy changes when they occur.

Use the **/MON:n** switch to specify the minimum number of changes that must occur before running again.

Use the **/MOT:m** switch to specify the minimum time, in minutes, that must elapse before running again.

If you only specify one of these switches, the other is assumed to have a value of 1.

When Monitoring is specified, Robocopy never terminates. Instead, it enters a loop that performs a normal Robocopy pass and then monitors the source directory tree for changes, until both **n** changes have been detected and **m** minutes have elapsed.

This whole process of copying and monitoring is repeated, ad infinitum, until the user manually terminates the Robocopy process.

For more information, see “Monitoring a Directory Tree” later in this document.

- Control bandwidth usage.

Use **/RH:hmm-hhmm** to specify Robocopy run hours.

When Run Hours have been specified, Robocopy will only start new copies when the local system time is between the two specified times.

Use the **/PF** switch to tell Robocopy to check its run hours on a Per File basis.

By default Robocopy will check its run hours on a per pass basis.

Use the **/IPG:n** switch to specify an Inter-Packet Gap of n milliseconds.

Robocopy will then sleep for the given time interval between each 64k chunk of file data transferred. This can be used to free up some bandwidth on slow links.

For more information, see “Controlling Bandwidth Usage” later in this document.

- Exercise more granular control over what is copied,
- Copy NTFS File Ownership Information, and
- Copy NTFS File Auditing Information.

Use the **/COPY:[D][A][T][S][O][U]** switch to specify exactly what to copy.

The flags listed copy file Data, Atttributes, Timestamps, NTFS Security (ACLs), NTFS Ownership information, and NTFS auditing information, respectively.

Some previous versions of Robocopy were able to copy NTFS ACLs. This version adds the ability to copy NTFS Ownership and Auditing information as well.

Use the **/COPYALL** switch to copy everything (equivalent to **/COPY:DATSOU**).

Use the **/NOCOPY** switch to copy nothing (this can be useful with **/PURGE**).

The default is **/COPY:DAT**, or **/COPY:DATS** if **/SEC** is specified.

Note If file Data is copied, then file Timestamps are also copied.

Note The **/TIMFIX** or **/SECFIX** switches are no longer supported. If you have scripts that use these switches you must edit them to use **/COPY:** instead. **/COPY:T** replaces the old **/TIMFIX**, and **/COPY:S** replaces the old **/SECFIX**.

Note You must have the Manage Audit Logs privilege to copy file auditing information using **/COPYALL** or **/COPY:U**.

- Control what is written to the output log.

Use the **/TS** (Time Stamps) switch to include source file timestamps in the output.

Use the **/FP** (Full Paths) switch to include full file pathnames in the output.

Use the **/NC** (No Classes) switch to remove Robocopy file classes from the output.

Use the **/NS** (No Sizes) switch to remove file and directory sizes from the output.

Use the **/NFL** (No File List) switch to suppress logging of file names.

Use the **/NDL** (No Directory List) switch to suppress logging of directory names.

Use the **/NJH** (No Job Header) switch to suppress output of the job header.

Use the **/NJS** (No Job Summary) switch to suppress output of the job summary.

Note File copy failures will still be logged even if **/NFL** is used.

Note If **/NDL** is used on its own, log files will contain fewer lines, but may use more disk space, as full pathnames of files will be logged instead of simple names.

- Log output to both the console window and a log file on disk, simultaneously.

Use **/TEE** to request simultaneous output to both the screen and also a log file specified by **/LOG** or **/LOG+**. This enables you visually monitor progress in a small console window on screen, while the full log is also written to disk.

Command-line Reference

To run Robocopy, use the following syntax at the command prompt:

ROBOCOPY *source destination [file [file]...] [options]*

The following table defines these syntax elements.

Variable	Meaning	Comments
<i>source</i>	Source directory	You can use <i>drive:\path</i> or <u><i>\\server\share\path</i></u>
<i>destination</i>	Destination directory	You can use <i>drive:\path</i> or <u><i>\\server\share\path</i></u>
<i>file</i>	Names of files to act upon	You can use wildcard characters (? and *). If no files are listed, Robocopy defaults to all files (*.*)
<i>options</i>	Command-line options you wish to use	Available options are described later in this document.

Tip To view brief usage instructions at the command prompt, run **ROBOCOPY** without specifying any command-line options.

The following table defines the command-line options that you can use with Robocopy.

Switch	Function Performed
/S	Copies subdirectories (excluding empty ones).
/E	Copies all subdirectories (including empty ones).
/LEV:n	Copies only the top <i>n</i> levels of the source directory tree.
/Z	Copies files in restartable mode (that is, restarts the copy process from the point of failure).
/B	Copies files in Backup mode (Backup copies are not restartable, but can copy some files that restartable mode cannot).
/ZB	Tries to copy files in restartable mode, but if that fails with an "Access Denied" error, switches automatically to Backup mode.

/COPY:copyflags	<p>Copies the file information specified by copyflags, which can be any combination of the following :</p> <p>D – file Data. S – file Security (NTFS ACLs). A – file Attributes. O – file Ownership information. T – file Timestamps. U – file auditing information.</p> <p>Source and destination volumes must both be NTFS to copy Security, Ownership or Auditing information.</p> <p>If no this switch is omitted the default is /COPY:DAT, or /COPY:DATS if /SEC is specified..</p>
/COPYALL	Copies Everything. Equivalent to /COPY:DATSOU .
/NOCOPY	Copies Nothing. Can be useful with /PURGE .
/SEC	Copies NTFS security information. (Source and destination volumes must both be NTFS). Equivalent to /COPY:DATS .
/MOV	Moves files (that is, deletes source files after copying).
/MOVE	Moves files and directories (that is, deletes source files and directories after copying).
/PURGE	Deletes destination files and directories that no longer exist in the source.
/MIR	Mirrors a directory tree (equivalent to running both /E and /PURGE).
/A+:{R A S H N T}	<p>Sets the specified attributes in copied files.</p> <p>The following attributes can be set:</p> <p>R – Read only S – System N – Not content indexed A – Archive H – Hidden T – Temporary</p>
/A-:{R A S H N T}	<p>Turns off the specified attributes in copied files.</p> <p>The following attributes can be turned off:</p> <p>R – Read only S – System N – Not content indexed A – Archive H – Hidden T – Temporary</p>
/CREATE	Creates a directory tree structure containing zero-length files only (that is, no file data is copied).
/FAT	Creates destination files using only 8.3 FAT file names.
/FFT	Assume FAT File Times (2-second granularity). Useful for copying to third-party systems that declare a volume to be NTFS but only implement file times with a 2-second granularity.
/MON:n	Monitors the source directory for changes, and runs again when n changes have been detected, and the minimum time interval specified by /MOT has elapsed.

/MOT:<i>n</i>	Monitors the source directory for changes, and runs again when a further <i>n</i> minutes have elapsed, and the minimum number of changes specified by /MON have been detected.
/RH:<i>hhmm-hhmm</i>	Defines the time slot during which starting new copies is allowed. Useful for restricting copies to certain times of the day. Both values must be 24-hour times in the range 0000 to 2359.
/PF	Makes more frequent checks to see if starting new copies is allowed (per file rather than per pass). Useful in stopping copy activity more promptly at the end of the run hours time slot.
/IPG:<i>n</i>	Inserts a delay of <i>n</i> milliseconds after each 64k chunk of file data is copied. Useful for freeing up bandwidth on slow lines.
/IA:{R A S H C N E T O}	Includes files with the specified attributes. The following file attributes can be acted upon: R – Read only A – Archive S – System H – Hidden C – Compressed N – Not content indexed E – Encrypted T – Temporary O - Offline
/XA:{R A S H C N E T O}	Excludes files with the specified attributes. The following file attributes can be acted upon: R – Read only A – Archive S – System H – Hidden C – Compressed N – Not content indexed E – Encrypted T – Temporary O - Offline
/A	Copies only files with the archive attribute set.
/M	Copies only files with the archive attribute set and then resets (turns off) the archive attribute in the source files.
/XJ	Excludes Junction points.
/XF <i>file</i> [<i>file</i>]	Excludes files with the specified names, paths, or wildcard characters.
/XD <i>dir</i> [<i>dir</i>]	Excludes directories with the specified names, paths, or wildcard characters.
/XC	Excludes files tagged as “Changed”.
/XN	Excludes files tagged as “Newer”.
/XO	Excludes files tagged as “Older”..
/XX	Excludes files and directories tagged as “Extra”.
/XL	Excludes files and directories tagged as “Lonely”.
/IS	Includes files tagged as “Same”.

/IT	Includes files tagged as "Tweaked".
/MAX:n	Excludes files larger than <i>n</i> bytes.
/MIN:n	Excludes files smaller than <i>n</i> bytes.
/MAXAGE:n	Excludes files with a Last Modified Date older than <i>n</i> days or specified date. If <i>n</i> is less than 1900, then <i>n</i> is expressed in days. Otherwise, <i>n</i> is a date expressed as YYYYMMDD.
/MINAGE:n	Excludes files with a Last Modified Date newer than <i>n</i> days or specified date. If <i>n</i> is less than 1900, then <i>n</i> is expressed in days. Otherwise, <i>n</i> is a date expressed as YYYYMMDD.
/MAXLAD:n	Excludes files with a Last Access Date older than <i>n</i> days or specified date. If <i>n</i> is less than 1900, then <i>n</i> is expressed in days. Otherwise, <i>n</i> is a date expressed as YYYYMMDD.
/MINLAD:n	Excludes files with a Last Access Date newer than <i>n</i> days or specified date. If <i>n</i> is less than 1900, then <i>n</i> is expressed in days. Otherwise, <i>n</i> is a date expressed as YYYYMMDD.
/R:n	Specifies the number of retries on failed copies. (The default is 1 million.)
/W:n	Specifies the wait time between retries. (The default is 30 seconds.)
/REG	Saves /R:n and /W:n in the registry as default settings.
/TBD	Waits for share names to be defined on a "Network Name Not Found" error.
/L	Lists files without copying, deleting, or applying a time stamp to any files.
/X	Reports all files tagged as "Extra"(including files not selected).
/V	Produces verbose output (including skipped files).
/TS	Displays source file timestamps in the output log.
/FP	Displays full pathnames of files in the output log.
/NC	Suppresses output of Robocopy file classes.
/NS	Suppresses output of file and directory sizes.
/NDL	Turns off logging of directory names. Full file pathnames (as opposed to simple file names) will be shown if /NDL is used.
/NFL	Turns off logging of file names. File names are still shown, however, if file copy errors occur.
/NJH	Turns off logging of the job header.
/NJS	Turns off logging of the job summary.

/NP	Turns off copy progress indicator (% copied).
/ETA	Shows estimated time of completion for copied files.
/LOG:file	Redirects output to the specified file, overwriting the file if it already exists.
/LOG+:file	Redirects output to the specified file, appending it to the file if it already exists.
/TEE	Displays output in the console window, in addition to directing it to the log file specified by /LOG or /LOG+.
/JOB:job	Reads parameters from the named Job file.
/SAVE:job	Writes the current parameter settings to the named Job file.
/QUIT	Quits after processing the command line. No files will be copied. Use /QUIT with /JOB to view job file contents.
/NOSD	Declares that no source directory is specified. Useful in template Jobs for which the source is provided at run time.
/NODD	Declares that no destination directory is specified. Useful in template Jobs for which the destination is provided at run time.
/IF	Includes files with the specified names, paths, or wildcard characters. Intended for use in Job files only.
/SD:path	Explicitly specifies the source directory for the copy. Intended for use in Job files only.
/DD:path	Explicitly specifies the destination directory for the copy. Intended for use in Job files only.

Usage Notes

The following sections provide more details about using the command-line options to tailor Robocopy to match your needs.

Backwards Compatibility

This version of Robocopy was developed and tested under Microsoft Windows XP.

The recommended minimum operating system platform for this version of Robocopy is therefore Microsoft Windows XP, or Microsoft Windows Server 2003.

Robocopy uses only standard, publicly documented Win32 APIs throughout. For example, the program uses CopyFileEx() to copy files, and FindFirstFile() and FindNextFile() to read directories. In theory, Robocopy should run on any operating system that supports these APIs, such as Windows NT version 4.0 and Windows 2000.

However if you choose to run this version of Robocopy on Windows NT version 4.0 or Windows 2000, please be aware that only limited testing was performed on these platforms, and Robocopy's theoretical performance may differ from practical results.

Note As Robocopy uses the CopyFileEx() Win32 API, it can not run on Windows NT 3.1, Windows NT 3.5x, nor Windows 95, Windows 98 or Windows ME.

Using Robocopy Within a UNIX Shell

You can specify all Robocopy switches in UNIX style (for example, **-ETA** instead of **/ETA**). You can also specify source and destination directory paths using the UNIX delimiter (/), rather than the native Windows delimiter (\). The only restriction is that any argument that begins with a UNIX slash mark (/) is treated as a switch if the argument contains only a single slash mark (/). In other words, */dir* is treated as a switch; but *//server/share/dir* and */download/test* are treated as paths.

This avoids any possible confusion between switches and single-level paths subordinate to the root of a drive. To specify a directory as an argument, you must use an alternate expression for its path, such as **X:/dir** or **//server/C\$/dir**.

Processing a Directory Tree

By default, Robocopy processes only the single source directory specified on the command line. To process the entire directory tree, including

subdirectories, use **/S** or **/E**. To limit the copy to only the top *n* levels of the source directory tree, use **/LEV:n**.

For example **/LEV:2** will cause Robocopy to process the top two directory levels only, i.e. just the root directory of the source and any directory directly under the root. Subdirectories of directories under the root will not be skipped. By default Robocopy will handle file and directory path names up to almost 32,000 characters in length. If for any reason you wish to disable this support for very long path names, use the **/256** switch. This causes Robocopy to revert to normal path name semantics, and a maximum path name length of 256 characters.

If the **/256** switch is used and Robocopy encounters a path name longer than 256 characters, one of the following errors may be reported, depending on the operation being performed on the very long path name at the time :

- The filename, directory name, or volume label syntax is incorrect.

- The system cannot find the file specified.

- The file name or extension is too long.

While walking a directory tree, Robocopy may encounter Junctions in the file system. These may be Volume Mount Points created using the MOUNTVOL command, or Directory Links created using the LINKD command. Robocopy handles Junctions in the source by creating a normal Directory of the same name in the destination, because it may not be possible to replicate the Junction in the destination. This is often the case if the source and destination are on different machines, as the target volume or directory of the Junction may not exist on the destination machine. Note that copying Junctions in this way can result in unexpectedly large volumes of data being copied if, for example, the source directory tree contains a Volume Mount Point. You can disable processing of Junctions by specifying the **/XJ** switch, which causes Robocopy to exclude Junctions from further processing. If **/XJ** is specified, Junctions are skipped.

Retrying After a Failed Copy

When a copy operation fails and returns an error, Robocopy waits and then retries the operation until it succeeds or until the retry limit is reached. The defaults are a 30-second wait and up to 1 million retries.

Changing Default Retry Parameters

Use the **/W:n** switch to change the wait time (in seconds) between retries, where *n* is a positive decimal integer, or zero (0) if you do not want any wait time between retries. Similarly, use the **/R:n** switch to change the retry limit. If an invalid value is given for **/W** or **/R**, Robocopy uses the default value.

To change the default retry parameters, use the **/REG** switch on a valid Robocopy command that specifies non-default values for **/W** and **/R**. When **/REG** is used, the values you specify for **/W** and **/R** are stored in the registry and used as default values for **/W** and **/R** in future Robocopy runs where **/W** and **/R** are *not* specified on the command line. You can always specify **/W** or **/R** to override the stored settings.

Handling “Network Name Not Found” Errors

Robocopy does not automatically retry all failed operations. Certain errors in some operations are not retried, where practical experience indicates that this would be futile. For example, the “Network Name Not Found” error usually indicates that a remote computer exists but does not have a share with the given name. Because manual intervention is required to create a suitable share name on the remote computer, this error usually causes the attempted operation to fail without a retry.

In some situations, however, this might not be the desired action. For example, in a software publishing environment it is common to delete a share name, update the contents of the share, and then recreate the share name. In this scenario, you can use the **/TBD** switch to retry a copy operation when Robocopy encounters a “Network Name Not Found” error.

Restarting from the Point of Failure

Normally Robocopy will restart failed copies from the beginning of the file. You can override this default behaviour by specifying the **/Z** switch (or the **/ZB** switch) which requests restartable copies - with **/Z**, failed copies will usually restart from close to the point of failure rather than the beginning of the file. See “Copying NTFS Security Information” later in this document for more information on **/ZB**.

Note If you use **/Z**, **/B**, or **/ZB**, this can decrease Robocopy performance and throughput significantly, as these options involve extra overhead. These options are therefore only recommended when experience indicates you really need them.

Specifying File and Directory Names

To specify a directory or file name or path name that includes one or more spaces, you must enclose the entire file or path name in quotation marks. For example:

```
ROBOCOPY "\\sales\reports\Fiscal 97" "C:\finance\Year End\sales" /XF "Budget 96*.**"
```

Note Any file names that appear in the command line must be specified as individual arguments separated from other arguments by white space, and not appended to the source or destination directory path names. If no such arguments appear in the command line, Robocopy defaults to all files.

By default, Robocopy treats all non-switch command-line arguments as file names, paths, or wildcard characters. These names can be alternated with switch arguments, but the command line is easier to read when they are grouped together.

The command line is parsed from left to right. The meaning of any non-switch argument depends on where it appears with respect to an **/XF** or **/XD** switch.

Normally there must be two non-switch arguments before any **/XF** or **/XD** switch, and these are treated as the path names of the source and destination directories, respectively. The only exception is when **/NOSD** or **/NODD** is used as a directory placeholder to define a Template Job File. See “Robocopy Job Files” later in this document for more details.

After the source and destination path names, and until Robocopy encounters an **/XF** or **/XD** switch, any non-switch argument is treated as either a file name or a wildcard character (but not a path) indicating one or more files to include as candidates for copying.

The **/XF** switch specifies that all subsequent file names, paths, and wildcard characters are files to exclude from copying until an **/XD** or **/IF** switch is found. The **/XD** switch specifies that all subsequent file names, paths, and wildcard characters are directories to exclude from copying until an **/XF** or **/IF** switch is found.

The **/IF** switch specifies that all subsequent file names and wildcard characters are candidates for copying until an **/XD** or **/XF** switch is found. **/IF** is intended for use within Robocopy Job Files, but it can also be used on the command line.

The following table compares these subtle differences in allowed values:

Argument	Names	Paths	Wildcard Characters
Source directory	Yes	Yes	No
Destination directory	Yes	Yes	No
Files to include	Yes	No	Yes
Files to exclude	Yes	Yes	Yes
Directories to exclude	Yes	Yes	Yes

The following example shows the correct use of these switches:

```
ROBOCOPY c:\source d:\dest *.c* /XF *.cpp /S /XD bin c:\source\unwanted
```

In this example, Robocopy recursively copies the directory tree (including subdirectories) with the root c:\source, excluding the subdirectories bin and c:\source\unwanted. All files whose extensions begin with c are copied, except files with the extension .cpp.

Using Robocopy File Classes

For each directory processed, Robocopy constructs a list of files in both the source and destination directories. This list matches the files specified on the command line for copying.

Robocopy then cross-references the lists, determining where files exist and comparing file times and sizes. The program places each selected file in one of the following classes.

File class	In source	In destination	Source/dest file times	Source/dest file sizes	Source/dest attributes
Lonely	Yes	No	n/a	n/a	n/a
Tweaked	Yes	Yes	Equal	Equal	Different
Same	Yes	Yes	Equal	Equal	Equal
Changed	Yes	Yes	Equal	Different	n/a
Newer	Yes	Yes	Source > destination	n/a	n/a
Older	Yes	Yes	Source < destination	n/a	n/a
Extra	No	Yes	n/a	n/a	n/a
Mismatched	Yes (file)	Yes (directory)	n/a	n/a	n/a

By default, Changed, Newer, and Older files are candidates for copying (subject to further filtering, as described later). Same files are not copied. Extra and Mismatched files and directories are only reported in the output log. Normally, Tweaked files are neither identified nor copied – they are usually identified as Same files by default. Only when /IT is used will the distinction between Same and Tweaked files be made, and only then will Tweaked files be copied.

Use the following switches to override this default behavior.

Switch	Function Performed
/XL	Excludes Lonely files and directories.
/IT	Includes Tweaked files.
/IS	Includes Same files.
/XC	Excludes Changed files.
/XN	Excludes Newer files.
/XO	Excludes Older files.
/XX	Suppresses reporting and processing of Extra files.

Note You do not need to specify any of these arguments to include the current version of source files in the destination tree. In most situations, Robocopy includes these files by default.

The **/XL** switch limits Robocopy to copying a file from the source directory tree only when a corresponding file of the same name already exists in the destination. This provides a convenient mechanism for maintaining a copy of a selected subset of the source tree. Using the **/IS** switch alone, with no other switches, forces a total refresh of the destination tree. Using the **/IT** switch provides a way to copy some Same files but not others by changing the

attributes of those you wish to be copied in either the source or the destination.

During a copy operation Robocopy places a January 1980 time stamp on the destination file, and updates this to the source file's time stamp only when the copy is complete. If you quit Robocopy during a copy operation, any incompletely copied destination file will therefore have an earlier time stamp than its corresponding source file. If you restart the same copy operation, Robocopy treats this file as an Newer file and will therefore complete the interrupted copy, unless you have specified /XN.

The most appropriate use for **/XO** is to synchronize two directory trees so that they can be updated simultaneously in separate areas. To ensure that the latest files are present in both directory trees, copy with **/XO** first in one direction and then in the other.

You can use switches to exclude files based on their size or time stamp.

/MAX:n excludes files larger than *n* bytes, and **/MIN:n** excludes files smaller than *n* bytes. Similarly, **/MAXAGE:n** excludes files with a last modified date older than *n* and **/MINAGE:n** excludes files with a last modified date newer than *n*, where *n* is either a time span in days (if *n* < 1900), or a date in the form YYYYMMDD (if *n* >= 1900).

Similarly, **/MAXLAD:n** excludes files with a Last Accessed Date older than *n* and **/MINLAD:n** excludes files with a Last Accessed Date newer than *n*, where *n* is either a time span in days (if *n* < 1900), or a date in the form YYYYMMDD (if *n* >= 1900).

Using Precise File Times and Long File Names

Although it is preferable to have both the source and destination directories reside on NTFS partitions, you can use Robocopy to copy to other file systems, such as HPFS or FAT. If you do so, however, file times might be inaccurate due to rounding errors, and long file names might not be preserved.

Handling FAT File Times on NTFS Drives

File-time granularity is 100 nanoseconds on NTFS, and two seconds on FAT, so copying a file from NTFS to FAT causes file times to be rounded to a value that the FAT file system can manage. When both the source and destination volumes are NTFS, file times are compared exactly. When either the source or destination volume is not an NTFS volume, Robocopy considers file times to be identical if they are within two seconds of each other. Without this two-second margin, the program might classify many otherwise unmodified files as Older or Newer files, which would result in unnecessary copying of unchanged files. Sometimes you need to override this handling of file times. Suppose you copy an NTFS tree to a FAT volume, and then copy the FAT tree (with its rounded file times) to a local NTFS drive. Later, when you want to recreate the original tree exactly, you do not want to refresh the entire tree.

In this scenario, you can use the **/IS** and **/COPY:T** switches together to copy just file times independently of file data for Same files, rather than copying the

files themselves again. See “Selectively Copying File Data” later in this document for more information on **/COPY**.

Some third-party operating systems declare that their volumes are NTFS, but only implement FAT-style file times with a 2-second granularity. When copying to such a destination from a true NTFS volume, file time rounding may occur, along with unnecessary copying of file data in subsequent jobs. If you encounter this situation, you can use the **/FFT** (FAT File Times) switch to tell Robocopy to consider file times to be identical if they are within 2 seconds of each other, even though both source and destination appear to be NTFS. This eliminates unnecessary copying in this scenario.

Handling Long File Names on FAT or HPFS Volumes

When the destination is on a FAT or HPFS volume, you might experience problems copying files and directories with long names or with names that include extended Unicode characters. To overcome these problems, use the **/FAT** switch. This switch prompts Robocopy to create destination files with system-generated names in the standard 8.3 FAT file system format.

Working with File Attributes

By default, Robocopy ignores most source file attributes (read-only, archive, system, and hidden) when selecting files to copy. Any file matching other specified conditions is copied regardless of its attribute settings. You can use command-line switches to modify this default behavior, as described in the following sections.

Including or Excluding Files Based on Specified Attributes

The **/IA:{R|A|S|H|C|N|E|T|O}** switch includes only files with one or more of the specified attributes set. Files that do not have any of the specified attributes set are excluded from further processing and are not copied.

/IA can be useful in data staging applications. For example, you can specify **/IA:R** to copy only read-only files from a test Web server to a live Web server. Developers can remove the read-only attribute from files on the test server while they edit and test the files. Until the read-only attribute is turned on again, these files are not replicated on the live server.

Another switch, **/XA:{R|A|S|H|C|N|E|T|O}**, can be used to exclude files if one or more of the specified attributes is set. For example, you can specify **/XA:R** to prevent read-only files from being copied. Similarly, **/XA:SH** excludes files with either the system or hidden attribute set.

Any combination of the following file attribute types can be used with **/IA** and **/XA**:

- R** – Read only files
- A** – Archive files
- S** – System files
- H** – Hidden files

- C** – Compressed files
- N** – files Not content indexed
- E** – Encrypted files
- T** – Temporary files
- O** – Offline files

Note Robocopy processes **/IA** before **/XA** when both switches are specified in the same Robocopy command.

The **/A** and **/M** switches allow only those source files with the archive attribute set to be selected for copying. After copying, the archive attribute of the source file is either left set (if **/A** was used) or turned off (if **/M** was used).

Note Temporary files are normally deleted by the owning application when it is no longer needed. When present on the system they are usually in use, and therefore copy delays or failures will often occur if you try to copy them.

Note Offline files have been physically removed to offline storage, and severe delays may be encountered in copying them, while the system retrieves them.

Changing File Attributes After Copying

After a file has been successfully copied to the destination, the attributes of the destination file are set by default to match those of the source file. To modify this default behavior, you can use two other switches:

/A+:{R|A|S|H|N|T} sets attributes, and

/A-:{R|A|S|H|N|T} resets or turns off attributes.

For example, **/A+:A** turns on the archive attribute in destination files (to make it easier to back them up). **/A-:RSH** removes the read-only, system, and hidden attributes from destination files (to make it possible to see and edit them).

The order of attribute operations on newly copied destination files is as follows:

1. Attributes are copied to the destination file from the source file.
2. Attributes specified by **/A+** are set (turned on).
3. Attributes specified by **/A-** are reset (turned off).

Any combination of the following file attribute types can be used with **/A+** and **/A-**:

- R** – Read only files
- A** – Archive files
- S** – System files
- H** – Hidden files

- N** – files Not content indexed
- T** – **Temporary files**

Maintaining True Replication

If you require the destination directory tree to be maintained as an *exact* mirror of the source directory tree, you must delete files and directories from the destination whenever they disappear from the source. To accomplish this, use the **/E** and **/PURGE** switches in combination, or use the **/MIR** switch. The **/PURGE** switch causes Robocopy to delete all Extra and Mismatched destination files and directories. After a Mismatched destination entry has been deleted, the corresponding source entry is then treated as a Lonely file or directory and processed accordingly. When **/PURGE** results in the deletion of an Extra or Mismatched destination directory, the entire directory tree, including all subordinate files and subdirectories, is deleted (even if **/S** or **/E** is not specified).

Caution Use **/PURGE** and **/MIR** very carefully. If you inadvertently specify **/PURGE** or **/MIR** with an existing destination directory, Robocopy can quickly delete a large amount of data from the destination.

To prevent specific directory trees or files from being deleted when you use **/PURGE**, you must exclude them by name, using the **/XD** or **/XF** options. **/XD** and **/XF** exclude files and directories before all other processing.

Note Because the **/XX** switch excludes Extra files from further processing, the **/PURGE** switch has no effect when **/XX** is also used.

Selectively Copying File Data

Once a file has been selected for copying, you can choose precisely which data associated with the file is copied using the **/COPY:copyflags** command line argument, where **copyflags** can be any combination of the following :

- **D** to copy file Data
- **A** to copy file Attributes
- **T** to copy file Timestamps
- **S** to copy NTFS Security information (ACLs)
- **O** to copy NTFS Ownership information
- **U** to copy NTFS aUditing information

For ease of use the following alternative switches are also provided :

- **/COPYALL** copies everything (equivalent to **/COPY:DATSOU**).
- **/NOCOPY** copies nothing (which can be useful if **/PURGE** is also used).
- **/SEC** copies data, attributes, and NTFS ACLs (like **/COPY:DATS**).

Any combination of these flags can be used, if you have the appropriate access rights. However, if file data is copied, file timestamps are also copied. By default Robocopy assumes **/COPY:DAT**, and will copy file data, attributes, and timestamps.

Note

You must have the Manage Audit Logs privilege to copy file auditing information using **/COPYALL** or **/COPY:U**.

To refresh security information for existing destination files and directories without copying file data, use the **/IS** switch together with the **/COPY** switch without the **D** flag. For example **/IS /COPY:SOU** would refresh all security information for all selected files, without copying any file data.

To refresh time stamps on existing destination files and directories without copying file data, use the **/IS** switch together with the **/COPY** switch without the **D** flag. For example **/IS /COPY:T** would refresh all security information for all selected files, without copying any file data.

Copying NTFS Security Information

If both the source and destination directories reside on NTFS volumes, you can use Robocopy to copy NTFS security information. The most common requirement is to preserve NTFS ACLs during a copy. To do this, use the **/SEC** switch to copy NTFS file permissions when you copy files to a destination for the first time. If you also need to preserve NTFS ownership and auditing information you should use **/COPYALL**.

Note If either the source or the destination volume is not NTFS, requests to copy security information are ignored. Only one attempt is made to copy NTFS security information for each file or directory copied.

To refresh security information for existing destination files and directories without copying file data, use the **/IS** switch together with the **/COPY** switch without the **D** flag. For example **/IS /COPY:SOU** would refresh all security information for all selected files, without copying any file data. See “Selectively Copying File Data” earlier in this document for more information on **/COPY**. When you copy security information, you might notice a short delay after Robocopy prints its banner and before it displays arguments on the screen. This delay occurs because the program is checking for the NTFS file system at both ends. Copying NTFS file security information fails if you have insufficient access privileges at either the source or the destination. For more information about access privileges, see “Scheduling Robocopy Jobs” later in this document.

You can copy security information between computers in separate domains even if they do not have trust relationships. However, this applies security information that is unknown to the destination computer to the destination directory tree. When you view this information on the destination, user

accounts are shown as “Account Unknown.” If you copy these permissions back to a computer in the originating domain, they are interpreted correctly again.

If you copy NTFS security information (ACLs) along with file data, it is possible to copy files to which you have read access, but not write access. After such a file is copied once, and the ACLs are applied, you may find that to get an “Access Denied” error when you try to copy the file again.

In this situation you should use the **/B** or **/ZB** switch to copy the files in Backup Mode.

/B copies all files with backup semantics (Backup Mode). **/ZB** first attempts to copy files in restartable mode (for greater resiliency) but if that fails with an “Access Denied” error it automatically retries the copy using Backup Mode.

Note You must have the Backup Files and Restore Files privileges to copy files in Backup Mode using **/B** or **/ZB**.

Note Backup mode copies are not restartable, but they enable you to copy some files as a Backup Operator that you would not be able to copy as a normal user.

Note If you use **/Z**, **/B**, or **/ZB**, this can decrease Robocopy performance and throughput significantly, as these options involve extra overhead. These options are therefore only recommended when experience indicates you really need them.

Moving Files

In some situations, you may want to move files instead of copying them, especially if disk space is at a premium on your network. You can use the **/MOVE** switch to delete source files from the source directory tree after they have been successfully copied to the destination. **/MOVE** also deletes empty directories from the source directory tree.

Even with **/MOVE** specified, however, Robocopy deletes only those source files that it successfully copies to the destination. There is no guarantee that a skipped source file is identical to its corresponding destination file (even if the file times and sizes are identical) until immediately following a successful copy. For that reason, some files and directories (such as skipped Same files) might remain in the source tree even when **/MOVE** is specified. You must determine whether to delete the remaining entries, and do so manually.

Note To move files but still preserve the source directory tree structure, use **/MOV** instead of **/MOVE**.

Minimizing Directory Fragmentation

When you copy a directory tree for the first time, the files contained in each directory are copied into it as it is created. The directory entry for each file is created first, and then the file data is copied.

If the tree contains large directories, however, this sequence can cause fragmentation of the directory structure. When disk allocation for the directory itself is expanded, the new directory extent can be separated from the rest of the directory by large amounts of file data, which can reduce later file system performance significantly.

Use the **/CREATE** switch to minimize directory fragmentation when you copy a large tree for the first time. When **/CREATE** is specified, Robocopy creates only the directory tree and zero-length files as placeholders for the actual files. No file data is copied. Immediately following a Robocopy command containing **/CREATE**, run the same command again without **/CREATE** to copy the file data to the destination.

Robocopy does not use **/CREATE** by default, because pre-creating a directory tree and then copying file data takes more time than just copying the files. For directory trees

Note When using this technique, do not use **/XO**, because the zero-length files created by the initial **/CREATE** command are tagged as “Older” files on the second run.

Robocopy Job Files

Given the large number of command-line options available, it is easy to create Robocopy commands that are extremely long, and unwieldy to manipulate. Using Robocopy Job Files can greatly simplify matters.

Robocopy Job Files are simple text files containing one Robocopy parameter per line, that you can create, view, and edit manually using a text editor, or, more simply, you can use Robocopy itself for all of these tasks.

There are three basic switches that are used in Robocopy Job File processing. The **/SAVE:job** switch tells Robocopy to save its current set of parameters to the named file, the **/JOB:job** switch tells Robocopy to read parameters from the named file, and the **/QUIT** switch tells Robocopy to terminate after processing the command line.

Note Both **/SAVE** and **/JOB** expect the file to have a .RCJ suffix, and append a .RCJ suffix to the provided filename if it does not have one.

For example, you may use a Robocopy command to back up source files as follows :

```
ROBOCOPY C:\SRC D:\SRC /E *.C* *.H*
```


To encapsulate this command in a Robocopy Job File, you could do the following :

```
ROBOCOPY C:\SRC D:\SRC /E *.C* *.H* /SAVE:SBACKUP
```

This creates a Robocopy Job File named SBACKUP.RCJ in the current directory that encapsulates all the Robocopy parameters you specified prior to the /SAVE switch.

Note /SAVE is actioned as soon as it is encountered on the command line. Any arguments on the command line after /SAVE will not be saved.

Note /SAVE writes to the specified file immediately, with no warning if the target file already exists. Care should be taken that you do not overwrite existing files when using /SAVE to create a new Robocopy Job File.

To run the resulting Robocopy Job File at a later date, assuming the same working directory, the following command would suffice :

```
ROBOCOPY /JOB:SBACKUP
```

Actually, the command given above to create a Robocopy Job File might not be ideal, as after processing the /SAVE switch and creating the Robocopy Job File, Robocopy would continue on and run with the given parameters.

To just create the Robocopy Job File without running the job, you can use the /QUIT switch as follows :

```
ROBOCOPY C:\SRC D:\SRC /E *.C* *.H* /SAVE:SBACKUP /QUIT
```

To view the verbatim contents of the generated .RCJ file you can either load it into a text editor (e.g. by running NOTEPAD SBACKUP.RCJ) or display its contents in the console window (e.g. by running TYPE SBACKUP.RCJ).

Alternatively, you can get a more succinct idea of its contents by running :

```
ROBOCOPY /JOB:SBACKUP /QUIT
```

This causes Robocopy to load the file, display its parameters, and then quit.

To edit a job file, you can use /JOB /SAVE and /QUIT together as follows :

```
ROBOCOPY /JOB:SBACKUP /XF *.HLP /SAVE:SBACKUP /QUIT
```

This loads the named job, adds parameters to exclude .HLP files from the job, and updates it on disk.

Alternatively, you can edit job files using a text editor. The format of Robocopy Job Files is straightforward. They contain one Robocopy parameter per line (parameters containing colons, e.g. /LOG:logfile or /MAX:n, count as one parameter and must be specified on a single line, with no spaces around the colon). Leading white space is ignored, as is any text after two consecutive

colons, which is considered to be a comment. View a Robocopy-generated .RCJ file to examine the format in more detail.

Multiple Robocopy Job Files can be specified on the Robocopy command line. For example, you might have a large list of files to be excluded from copying. You could use an editor to create a text file containing this exclusion list as follows :

```
/XF
file1
file2
file3
...
```

Once this is saved with a .RCJ suffix, e.g. EXCLUDE.RCJ, it can be used in conjunction with other Robocopy Job Files, for example as follows :
ROBOCOPY /JOB:SBACKUP /JOB:EXCLUDE

Note When multiple job files are used, their contents are cumulative. For example if a job specifies **/XA:R** and is used in conjunction with another that specifies **/XA:SH** then the result will be **/XA:RSH**. This also means that you must make sure that only one source directory and only one destination directory is specified in any set of job files that are used together.

It is sometimes useful to create Robocopy Job Files that can be used as templates for similar types of copies, and for this purpose it is perfectly valid to save a job file without specifying a source directory or destination directory or both. Such a job file cannot be run on its own, of course, unless the source and destination directories are specified at run time.

For example:

```
ROBOCOPY *.c* *.h* /SAVE:SBACKUP /QUIT
```

Could be used create a template job SBACKUP.RCJ for backing up source files.

To use this template to back up different projects you might later run:

```
ROBOCOPY /JOB:SBACKUP C:\PROJECTS\PROG1 D:\PROJECTS\PROG1
ROBOCOPY /JOB:SBACKUP C:\PROJECTS\PROG2 D:\PROJECTS\PROG2
```

However, if the list of files you wish to include contains a specific file name (as opposed to wild cards like *.c*), such as MAKEFILE, then this approach will fail. You might attempt to create such a template job as follows:

```
ROBOCOPY *.c* *.h* MAKEFILE /SAVE:SBACKUP /QUIT
```

The problem here is that Robocopy assumes that the first three non-switch, non-wild card arguments are the source pathname, the destination pathname, and the first of a list of files to include, respectively, and therefore will assume that MAKEFILE is the source directory, rather than a filename to include.

To handle this situation you can use the **/NOSD** and **/NODD** switches in place of the source and destination directories respectively. In the example above you should use the **/NOSD** switch to declare that no Source Directory is being specified, and the **/NODD** switch to declare that no Destination Directory is being specified, as follows:

```
ROBOCOPY /NOSD /NODD *.c* *.h* MAKEFILE /SAVE:SBACKUP /QUIT
```

Obviously, both a source directory and a destination directory must be specified, so **/NOSD** and **/NODD** are only useful in the creation of template job files. However, template job files can be a powerful tool, as multiple job files can be specified on a single Robocopy command line.

Extending the example of backing up source files further, you could create a template job specifying file section options, then further jobs for specific source code projects:

```
ROBOCOPY /NOSD /NODD *.c* *.h* MAKEFILE /SAVE:SB /QUIT
ROBOCOPY C:\PROJECTS\PROG1 D:\PROJECTS\PROG1 /SAVE:P1 /QUIT
ROBOCOPY C:\PROJECTS\PROG2 D:\PROJECTS\PROG2 /SAVE:P2 /QUIT
Etc...
```

Such a job structure would enable you to back up your projects simply and consistently using brief commands like the following:

```
ROBOCOPY /JOB:SB /JOB:P1
ROBOCOPY /JOB:SB /JOB:P2
Etc...
```

This is just an introduction to Robocopy Job Files with simple and straightforward examples, but their flexibility and open-endedness enables large and complicated jobs to be created and managed with ease.

Scheduling Robocopy Jobs

You can create Robocopy jobs that run regularly and automatically in the background to maintain local mirrors of remote directory trees. In Windows 2000, use the **AT** command, or use the **SOON** command with the Windows 2000 Schedule service. (The **SOON** command runs *Soon.exe*, a *Windows® 2000 Resource Kit* tool named Near-Future Command Scheduler.) By default, the Schedule service logs on as the system account for the local system, which has no network access. Scheduled jobs run in the same context as the Schedule service. To run a scheduled Robocopy job, however, Robocopy must connect to network drives, which requires some additional configuration. The following sections describe two ways to accomplish this.

Using the Local System Account

You can leave the Schedule service running in the context of the local system account. You can then schedule batch files by using the following commands:

```
NET USE \\remoteserver\IPC$ /USER:userid password
ROBOCOPY \\remoteserver\sourcepath \\localserver\destpath ...
```

```
NET USE \\remoteserver\IPC$ /DEL
```

Within each job, credentials with remote servers are validated by connecting to the IPC\$ (Interprocess Communication) share on remote computers using an appropriate user account, rather than the local system account. The disadvantage of this approach is that user passwords are stored in the batch files; however, passwords can be protected by storing the batch files on an NTFS volume, and setting appropriate NTFS permissions on the batch files.

Using a User Account

You can configure the Schedule service to log on as a user account. Once the user account has been granted appropriate access to source and destination servers, you can schedule Robocopy jobs to copy files between them.

To configure the Schedule service to log on as a user account

1. In Services in Control Panel, click **Schedule**, and then click **Startup**.
2. In the **Startup Type** box, click **Automatic** or **Manual**.
3. In the **Log On As** box, select the account in which to run the service, and specify the correct logon password.

Tip Because drive mappings can be changed by users, it is more reliable to use UNC names for source and destination directories in scheduled Robocopy jobs. For example:

```
ROBOCOPY \\server1\share1\source\ \\server2\share2\dest ...
```

Note The **AT** command and the Schedule service are documented in Windows 2000 online Help. The **SOON** command is documented in the *Microsoft® Windows® 2000 Resource Kit*.

Monitoring a Directory Tree

Using the **/MON:n** and **/MOT:m** switches you can Monitor the source directory tree for changes, and copy changes as and when they occur. The **/MON:n** switch specifies the minimum number of changes that must occur before running again. The **/MOT:m** switch specifies the minimum time, in minutes, that must elapse before running again.

If you only specify one of these switches, the other is assumed to have a value of 1.

When Monitoring is specified, Robocopy never terminates. Instead, it enters a loop that performs a normal Robocopy pass and then monitors the source directory tree for changes, until both **n** changes have been detected and **m** minutes have elapsed. When both conditions are met, another Robocopy pass

is started. This whole process of copying and monitoring is repeated, ad infinitum, until the user manually terminates the Robocopy process.

As a minimum, Robocopy monitors the source directory tree for changes to file and directory names, sizes, and last write times. If attributes are to be copied Robocopy will also monitor the tree for attribute changes. Similarly, if security information is to be copied, Robocopy will also monitor the tree for security changes.

Consequently, a user performing an edit and saving a single new file, for example, may cause multiple changes of various types to be reported back to Robocopy. Therefore some experimentation may be required to determine appropriate threshold values for **/MON** and **/MOT** for each Robocopy job, especially those monitoring busy trees.

Controlling Bandwidth Usage

Using the **/RH:hhmm-hhmm** switch to specify Robocopy's Run Hours you can define a time slot in which starting new copies is allowed; outside of this time slot new copies will not be started, and Robocopy will sleep, waiting for the start time to come around.

Both times must be in 24-hour format, must be exactly 4 digits long in the range 0000 to 2359, and must be different so as to specify a minimum two-minute window in which new copies may be started.

For example **/RH:2200-0600** would specify that copies can be started when the local system time is between 10 p.m. and 6 a.m. (or, to be exact, between 10:00:00 p.m. and 06:00:59 a.m., as Robocopy only looks at the hours and minutes components of the local system time when deciding if starting new copies is allowed).

By default Robocopy will check to see if it is within the defined Run Hours time slot at the start of each Robocopy pass (i.e. before starting the initial copy pass; and, if **/MON** or **/MOT** were used, at the start of each subsequent copy pass).

You should use the **/PF** switch if you would like the Run Hours check to be performed before each individual file copy is started.

Once a copy is actually started it always runs to completion, and therefore it is probable that a copy pass or file copy that starts close to the end of the Run Hours time slot will not complete until some time after the end time specified. Experimentation may therefore be required to determine an appropriate end time for your needs.

The most obvious use of **/RH** is to limit Robocopy's copy activity to a time slot that is outside of normal business hours (e.g. **/RH:2200-0600**), to limit Robocopy's use of network bandwidth and improve network performance for interactive users during the day. This is particularly useful for Robocopy jobs that run for extended periods of time, for example those that process very large trees, or Monitoring jobs.

However, you can also use **/RH** to schedule a job to start at a later time. For example, you could start Robocopy in a command window with **/RH:1900-1901** just before you leave work to go home, and the job would start copying at 7pm, and run overnight.

On very slow links heavy Robocopy activity can easily consume large percentages to the available network bandwidth, and seriously impact network performance. In such a case you can use the **/IPG:n** switch to specify an Inter-Packet Gap of *n* milliseconds.

Robocopy usually copies files in 64 kilobyte packets with no delay between receiving a packet and requesting the next. When **/IPG** is used Robocopy will wait for the given time interval before requesting the next packet, thereby freeing up network bandwidth. Again, experimentation may be required to determine an appropriate value for **/IPG**.

Output Log

During a run, Robocopy creates an Output Log that comprises a Job Header detailing the command line arguments used for the job, a list of files and directories processed during the job, and a Job Summary that shows totals copied, etc. By default, Output Log is written to the command prompt. This output can be redirected to a file, either by using the standard command-line operators (> or >>), or by using one of the command-line options:

- **/LOG:file** redirects output to the specified file, overwriting the file if it already exists.
- **/LOG+:file** redirects output to the specified file, appending it to the file if it already exists.

You can also use **/TEE** in conjunction with one of the above options. This causes Robocopy to log its output to both the command prompt (for visual progress monitoring) and the specified file (for a permanent record of the Robocopy run).

Aside from error reporting and retries, Robocopy produces one line of output per source file or directory. Lines for directories show the number of files in that directory specified for copying (where known) and the full path of the directory. Lines for files indicate the name of the file, the size of the file, and what was done with the file.

One line is also generated for each Mismatched file and directory and each Extra file and directory in the destination. Lines for Extra files include the full path and name of the file.

By default, the only Extra files reported are those that match files specified for copying on the command line. In most cases, this is more efficient. For example, if you are refreshing CPP files, you probably do not need information about TXT files in the destination. If you want a list of all Extra files in the destination, regardless of their type, use the **/X** switch.

No output is produced for skipped files. To obtain a verbose listing that shows all source files specified for copying on the command line, including skipped files, use the **/V** switch.

Robocopy provides copy progress information (% copied) by default. You can use the **/NP** switch to suppress the display of progress information. This can be useful when output is redirected to a file.

To see the start time of each file copy and the estimated time of completion based on the observed throughput of previous copies, use the **/ETA** switch. Times are displayed after the file name in the format HH:MM → HH:MM (start → finish).

Finally, to obtain only a list of the files that would be copied (without actually copying them), use the **/L** switch.

This default contents of the output log can be modified using the following command line switches :

- **/TS** displays the file timestamps for every file processed.
- **/FP** replaces simple file names with full file pathnames in the output.
- **/NS** (No Sizes) suppresses output of file sizes.
- **/NC** (No Classes) suppresses output of file class "Text Tags" (see below).
- **/NDL** (No Directory List) suppresses output of the directory list. When **/NDL** is used, full pathnames of processed files are logged instead of the simple file name, so that problems can be pinpointed. Because of this, note that log files can actually increase in size if **/NDL** is used. Directories deleted by **/PURGE** (or which *would* be deleted if **/L** was omitted, if **/PURGE** and **/L** were both used) are always logged even if **/NDL** is used.
- **/NFL** (No File List) suppresses output of files processed. File copy failures are still logged even if **/NFL** is used, so that problems can be isolated. Files deleted by **/PURGE** (or which would be deleted if **/L** was omitted, if **/PURGE** and **/L** were both used) are always logged even if **/NFL** is used.
- **/NJH** (No Job Header) suppresses output of the job header.
- **/NJS** (No Job Summary) suppresses output of the job summary.

Combinations of these switches allow quite a bit of flexibility in terms of what is logged. For example, you could use **/NDL /NC /NS** to produce a list of just the full pathnames of files that were processed during the run, or you could use **/NDL /NFL** to produce just the summary report for the run (see "Run Summary" below).

Text Tags

Each line in the output log begins with a brief text tag, which is formatted according to the following rules:

- All capital letters indicate an anomaly that should be investigated.
- Initial capital letters indicate a file that was selected for copying.
- All lowercase letters indicate a file that was skipped (displayed only if the **/V** switch is used).

The text tags that indicate copying are left-aligned. Tags that indicate skipping are right-aligned. Tags that indicate anomalies are placed farther to the left than other tags. Error messages always start in the first column. This arrangement simplifies the task of scanning through even a verbose listing, and makes it easier to identify new downloads, anomalies, and network errors. If few files are copied, the left margin of the output log is mostly blank. Copied files and anomalies are easily spotted non-blank entries in the left margin of the output log.

Tags for Files

The following table lists the tags that apply to files.

File Tag	Meaning
*MISMATCH	The source file corresponds to a destination directory of the same name. The source file is skipped, and housekeeping of the destination tree is recommended.
*EXTRA File	The file exists in the destination but not the source (an Extra file). Some housekeeping of the destination tree may be required.
New File	The file exists in the source but not in the destination (a Lonely file). The file is copied; to skip this file, use /XL .
lonely	The file exists in the source but not in the destination (a Lonely file). The file is skipped; to copy this file, omit /XL .
Newer	The source file has a later time stamp than the destination file. The file is copied; to skip this file, use /XN .
newer	The source file has a later time stamp than the destination file. The file is skipped; to copy this file, omit /XN .
Older	The source file has an earlier time stamp than the destination file. The file is copied; to skip this file, use /XO .
older	The source file has an earlier time stamp than the destination file. The file is skipped; to copy this file, omit /XO .
Changed	The source and destination files have identical time stamps but different file sizes. The file is copied; to skip this file, use /XC .
changed	The source and destination files have identical time stamps but different file sizes. The file is skipped; to copy this file, omit /XC .
Tweaked	The source and destination files have identical time stamps and

	file sizes but different attributes. The file is copied because /IT has been specified; to skip this file, omit /IT .
Same	The source and destination files have identical time stamps and file sizes. The file is copied; to skip this file, omit /IS .
same	The source and destination files have identical time stamps and file sizes. The file is skipped; to copy this file, use /IS .
attrib	At least one source file attribute matches the attributes specified by /XA . The file is skipped; to copy this file, modify or omit /XA .
named	The file is skipped because it was named following an /XF switch. To process this file, amend the argument following /XF .
large	The file is skipped because it is larger than /MAX:n bytes. To process this file, increase the value of n .
small	The file is skipped because it is smaller than /MIN:n bytes. To process this file, reduce the value of /MIN:n .
too old	The file is skipped because it is older than the date defined by /MAXAGE:n or /MAXLAD:n . To process this file, amend n to specify an earlier date.
too new	The file is skipped because it is newer than the date defined by /MINAGE:n or /MINLAD:n . To process this file, amend n to specify a more recent date.

Tags for Directories

The following table lists the tags that apply to directories.

Directory Tag	Meaning
(blank)	A normal directory.
*MISMATCH	This source directory corresponds to a destination file of the same name and cannot be processed. Housekeeping of the destination is recommended.
*EXTRA Dir	The directory exists in the destination but not the source (an Extra directory). Some housekeeping of the destination tree may be required.
lonely	The directory exists in the source but not the destination (a Lonely directory). The directory is skipped; to process this directory, omit /XL .
named	The directory is skipped because it was named following an /XD switch. To process this directory, amend the argument following /XD .
junction	The entry is actually a junction, and is skipped because /XJ was specified. To process this junction and copy it as a directory, omit /XJ .

Run Summary

If you have not specified the **/NJS** (No Job Summary) switch, then before Robocopy stops, it displays a summary of its activities at the command prompt (or redirects the information to a file). This output summarizes the volume of data processed in the following format:

	Total	Copied	Skipped	Mismatch	FAILED	Extras
Dirs:	13	0	13	0	0	1
Files:	157	2	155	0	0	3
Bytes:	33.73 m	691.5 k	33.05 m	0	0	162
Times:	0:01:33	0:01:04			0:00:00	0:00:29
Speed:		10952 bps				
Speed:		0.626 mb/min				

The first column of the run summary indicates the total number of files and directories processed, the total size of source files matching the file names specified for copying, and the total time taken by the operation. The remaining columns provide a breakdown of the totals, as described in the following table.

Column	Information Provided
Copied	Number of directories created and files copied
Skipped	Number of directories processed but not created, and number of files skipped

Mismatch	Number of Mismatched files and directories
FAILED	Number of items not successfully processed within the retry limit
Extras	Number of items present in the destination but not the source

The Times line of the summary provides timing information for the run, as described in the following table. Times are displayed in HH:MM:SS format.

Column	Information Provided
Copied	Time spent copying files (excluding retry wait times)
FAILED	Time spent waiting between retries for failed operations
Extra	Time spent scanning directories and performing other functions

Note Large times in the FAILED column usually indicate network problems.

Finally, if files were copied, the Speed lines show throughput information, calculated by dividing the volume of data copied by the time spent copying. The same data is displayed twice: first in bytes per second (bps), and then in megabytes per minute (mb/min).

Return Code

The return code from Robocopy is a bit map, defined as follows:

Hex Bit Value	Decimal Value	Meaning If Set
0x10	16	Serious error. Robocopy did not copy any files. This is either a usage error or an error due to insufficient access privileges on the source or destination directories.
0x08	8	Some files or directories could not be copied (copy errors occurred and the retry limit was exceeded). Check these errors further.
0x04	4	Some Mismatched files or directories were detected. Examine the output log. Housekeeping is probably necessary.
0x02	2	Some Extra files or directories were detected. Examine the output log. Some housekeeping may be needed.
0x01	1	One or more files were copied successfully (that is, new files have arrived).
0x00	0	No errors occurred, and no copying was done. The source and destination directory trees are completely synchronized.

You can use this information in a batch file to report the most serious anomalies, as follows:

```
if errorlevel 16 echo ***FATAL ERROR*** & goto end
if errorlevel 8 echo **FAILED COPIES** & goto end
if errorlevel 4 echo *MISMATCHES* & goto end
if errorlevel 2 echo EXTRA FILES & goto end
if errorlevel 1 echo Copy successful & goto end
if errorlevel 0 echo --no change-- & goto end
:end
```

Alternatively, full details of the return code could be reported as follows:

```
if errorlevel 16 echo ***FATAL ERROR*** & goto end
if errorlevel 15 echo FAIL MISM XTRA COPY & goto end
if errorlevel 14 echo FAIL MISM XTRA & goto end
if errorlevel 13 echo FAIL MISM COPY & goto end
if errorlevel 12 echo FAIL MISM & goto end
if errorlevel 11 echo FAIL XTRA COPY & goto end
if errorlevel 10 echo FAIL XTRA & goto end
if errorlevel 9 echo FAIL COPY & goto end
if errorlevel 8 echo FAIL & goto end
if errorlevel 7 echo MISM XTRA COPY & goto end
if errorlevel 6 echo MISM XTRA & goto end
if errorlevel 5 echo MISM COPY & goto end
if errorlevel 4 echo MISM & goto end
if errorlevel 3 echo XTRA COPY & goto end
if errorlevel 2 echo XTRA & goto end
if errorlevel 1 echo COPY & goto end
if errorlevel 0 echo --no change-- & goto end
:end
```

Feedback

For questions or feedback concerning this tool, please contact rkinput@microsoft.com.

If you want to report a problem encountered when using Robocopy, please supply the following information.

- Version number
- Operating system and Service Pack for all computers
- File system
- Complete Robocopy command line or batch file
- Output log
- Directory listings
- Dr. Watson log (for application errors)

- Method of running Robocopy (from the command prompt, using Schedule service, and so on)
- User account information.

© 1985–2003 Microsoft Corporation. All rights reserved.