`tapecon` User Guide

Revision 1.0 July 2024

Copyright © 2023, 2024 by [avkl1m](#)

`tapecon` Introduction

`tapecon` is a Windows* command line application that allows to back up files to and restore from a tape drive. `tapecon` features include:

- Support of the popular USB-connectable tape drives, including Sony* AIT*, DAT 320 (DDS7), and DLT* VS1
- Flexible ways to specify files to back up: directly or with wildcards, from multiple directories, optionally including subdirectories
- Optional data integrity protection of backed-up files using hashing algorithms such as Crc32 or Crc64
- Incremental backups: backing up only the files that have changes since the last backup
- Support for multi-volume backups: using multiple tape volumes to accommodate large backup sets.

`tapecon` is built using `TapeNET` library for using tape drives under Microsoft* Windows and .NET*. `tapecon` can run under Microsoft Windows 10 and Windows 11.

`tapecon` and `TapeNET` are free, open-source software distributed under MIT license. Refer to the license file `LICENSE.txt` for more information.

CAUTION: When using `tapecon`, it's advisable to follow best backup practices, including employing multiple backup methods, not relying solely on this tool, and [verifying](#) or [validating](#) the backups regularly.

Why back up to tape drives?

The popular USB-connectable tape drives, such as Sony AIT, DAT 320 (DDS7), or DLT VS1, as well as the tape media for them (cassettes or cartridges), have become extremely inexpensive recently, due to low demand. Many potential users view them as outdated or even not supported anymore.

While portable hard drives and solid-state drives (SSDs) certainly exceed the tape drives in both capacity and data read-write rates, the tape drives still offer certain advantages:

- Lower cost per capacity, since the tape media prices for these drives have come down dramatically
- Multi-volume support for large backup sets, compensating for smaller capacity per single volume
- Long shelf life and WORM (write once, read many times) capability of tape media
- Virtually full protection against viruses and ransomware, since the malicious software cannot access even the loaded tape media, leave alone separately stored tape media
- Certain rustic charm of using tape technology – in a very modern software environment!

The popular USB-connectable tape drives, such as Sony AIT, DAT 320 (DDS7), or DLT VS1, are fully supported by Windows 10 and Windows 11, including driver availability through the standard Windows distribution and/or from Windows Update.

However, the selection of the backup software that could use the drives has been rather modest. Most contemporary backup applications either do not support tape drives at all, or only work with the expensive professional LTO* tape systems. `tapecon` closes this gap by providing a free, open-source backup application for tape drives.

`tapecon` quick start examples

[Backup](#) all files from the directory `C:\My Data` including subdirectories, appending the new backup set to the backup sets already on the tape, then verify the newly backed up files by comparing to the original ones:

```
tapecon -d:0 -subdir -backup "C:\My Data" -verify
```

Same as previous, but backup only the files that have changed or been newly added since the previous backup:

```
tapecon -d:0 -subdir -inc -backup "C:\My Data" -verify
```

[Restore](#) the most recent backup set to the directory `D:\Restored Files`, restoring the backed-up directory structure underneath and keeping the existing files (rename the restored files if necessary):

```
tapecon -d:0 -target "D:\Restored Files" -subdir -exist:keep -restore
```

[List](#) the content of the tape media:

```
tapecon -d:0 -list
```

[Validate](#) the second most recent backup set using the stored CRC values:

```
tapecon -d:0 -validate -1
```

How `tapecon` stores files on tape

`tapecon` stores every batch of files specified in a single [-backup task](#) as a **backup set**. A single tape media volume can contain multiple backup sets.
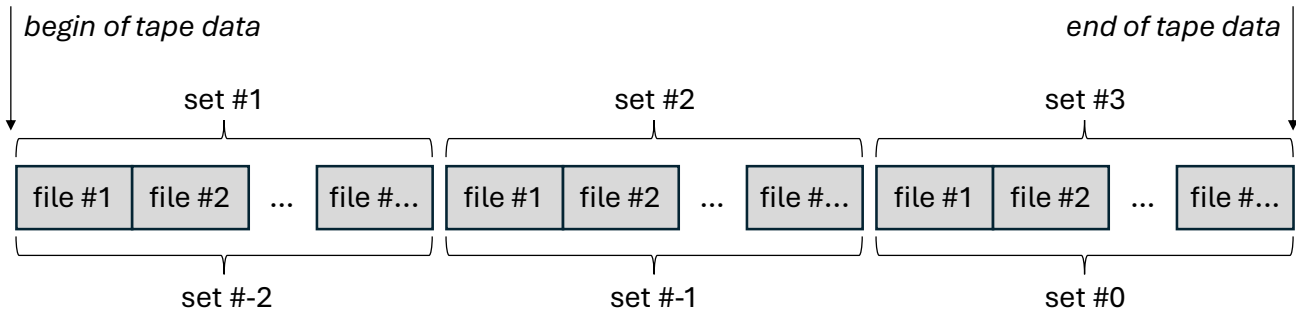
The information about the backup sets and the files they contain `tapecon` stores in a dedicated section of the tape called the **table of content** (**TOC**). Depending on the drive capabilities, `tapecon` can place the TOC in a dedicated tape partition or in the special last backup set. S. [-format task](#) for the discussion of TOC placements.

Backup set indexes

Each `tapecon` backup can be identified by either of the two **backup set indexes**:

- From the oldest (first recorded) to the latest (last recorded): 1, 2, 3, … These indexes are positive order, with #1 assigned to the oldest (first recorded) set, #2 to second oldest, etc.
- From the latest (last recorded) to the oldest (first recorded): 0, -1, -2, … These indexes are zero and negative numbers, with #0 assigned to the latest (last recorded) set, #1 to the second latest, etc.

The following illustrates backup set indexes for the case of three backup sets on the tape.

begin of tape data   *end of tape data*

| set #1 | set #2 | set #3 |

| file #1 | file #2 | ... | file #... | file #1 | file #2 | ... | file #... | file #1 | file #2 | ... | file #... |

| set #-2 | set #-1 | set #0 |

In the illustrated case, the latest (last written) backup set, as an example, can be identified by either index #3 or index #0.

Using the indexes from the end of tape data can be convenient to identify recently recorded backup sets, without the need to know their sequential number from the begin of the tape data. For example, the last recorded backup set can always be identified by the index #0, the second last by #-1, etc., regardless how many backup sets have been recorded on the tape.

As user input, `tapecon` can parse either of the index specifications, from the oldest or from the latest backup set. When outputting set indexes, `tapecon` provides both specifications.

Incremental backup sets

`tapecon` can be instructed to back up only the files that have changes since the last backup – that is, only those that have changed or have been newly added compared to the most recent backup set. `tapecon` will mark the newly created backup set, which contains the changed and newly added files, as an **incremental backup set**. The next backup can likewise be incremental and its backup set also marked as incremental etc.

When restoring files from an incremental backup set, `tapecon` will by default refer back to all previous incremental backup sets to identify and restore the most up-to-date version of every particular file. This behavior can be optionally disabled if a file version from a particular set is needed.

tapecon Command Line

`tapecon` is invoked using a command line.

Command Line Notation Conventions

This document uses the following notation conventions for the command line.

- **Square Brackets `[ ]`**: Indicate optional elements. The enclosed options can be included or omitted.
- **Angle Brackets `< >`**: Denote definable names or placeholders that should be replaced with actual values.
- **Vertical Bar `|`**: Represents a choice between multiple options. Only one of the options should be selected.
- **Double Periods `. .`** : Signify that the preceding element can be repeated multiple times.

Example:

As an example of using the conventions, here is the format for the `-restore` command line flag:

`-restore [<set_index>] [<files>..]`

In this example:

`-restore [<set_index>]`: The `-restore` flag can optionally be followed by a parameter `<set_index>`.

`[<files>..]`: One or more files can be specified, or this part can be omitted.

The following can be an example of specifying the `-restore` flag:

`-restore -2 *.doc* *.txt`

In this example, the parameter `<set_index>` has been specified with the value -2 and the files have been specified (using `*` wildcard) as `*.doc*` and `*.txt`.

Command Line Format

The `tapecon` command line consists of one or several flags each optionally followed by options. General command line format:

`tapecon -<flag1> [<option>..] [-<flag2> [<option>..] ..]`

Task Flags and Mode Flags

There are **task flags** and **mode flags**. The task flags direct `tapecon` to fulfill certain tasks, for example `-backup` to backup files or `-restore` to restore files.

The mode flags instruct `tapecon` how to fulfill the following tasks. For example, `-subdirectories` mode flag instructs `tapecon` to recurse subdirectories when performing the `-backup` task.

A mode flag may be used with several different task flags. For example, `-subdirectories` mode flag also instructs `tapecon` to restore files with subdirectories when performing the `-restore` task.

Flag Format

General flag format: `-<flag> [<option>..]`

Alternative formats to specify a single option (all equivalent): `-<flag> <option>`, `-<flag>:<option>`, or `-<flag>=<option>`

Flags and options are **not** case-sensitive. Therefore, using `-crc` mode flag as an example, `-crc:on`, `-crc:ON`, `-Crc:on`, and `-CRC:On` are all equal. The options that are file and directory (folder) names do preserve the case for file operations.

The flags and options are processed sequentially from left to right. Therefore, make sure to specify the mode flags related to a certain task before the flag for the task. For example, to execute a backup task of the directory `C:\ToBackup` recursing through its subdirectories, specify:

```
tapecon -subdirectories -backup C:\ToBackup
```

To perform several tasks in one invocation of `tapecon`, specify them sequentially. For example, to perform a backup, then to conduct a validation pass of the backed-up set, and finally to eject the tape, specify:

```
tapecon -subdirectories -backup C:\ToBackup -validate -eject
```

Specifying on/off mode flags

There are multiple equivalent ways to specify an on/off mode flags. Using `-crc` mode flag as an example:

- Switching a mode "**on**" (all equivalent): `-crc`, `-crc on`, `-crc=on`, `-crc:on`, `-crc=on`, `-crc+`
- Switching a mode "**off**" (all equivalent): `-crc off`, `-crc=off`, `-crc:off`, `-crc-`

Task Flags

The task flags direct `tapecon` to fulfill certain tasks. For example, the `-backup` task flag directs `tapecon` to backup files; the `-restore` task flag directs `tapecon` to restore files.

Drive Selection Task: `-drive <drive_number>`

Use the `-drive` task flag to specify the tape drive to open and use for the subsequent tasks.

- Alternative formats: `-d <drive_number>`

- Specify the `<drive_number>` parameter as an integer from `0` on. If no number is specified, `0` is assumed.

Include this flag before all other flags handling the drive.

The `-drive` task flag is only useful in conjunction with one or more task flags provided afterwards, to perform actions with the specified drive. Therefore, this flag is usually the first task flag provided on `tapecon` command line.

If the flag has not been specified before the first task flag, `tapecon` will attempt to open and use tape drive `0`.

Format Task: `-format [single]`

Use the `-format` task flag to format the tape media in the drive.

- Provide the `single` parameter to enforce formatting with a single partition and so placing the TOC in the special last backup set, instead of a dedicated partition.

CAUTION: Formatting a tape will irreversibly destroy all data previously recorded on the tape media!

By default, on the drives that support multiple partitions, the tape will be formatted with two partitions: for the content and for the table of content (TOC), as this provides the optimal performance and reliability.

To enforce formatting with a single partition and so placing the TOC in the special last backup set in the same partition as the content, specify `single` option. This might help decrease tape wear, at the expense of somewhat slower access to TOC.

Notice that some drives, such as DLT VS1 drives, do not require tape media formatting to start using a new tape media.

Examples:

Format the media in drive 0, providing for the TOC placement in a dedicated partition, if the drive supports multiple partitions:

```
tapecon -d:0 -format
```

Format the media in drive 0, providing for the TOC placement in the special last backup set in the same partition as the content:

```
tapecon -d:0 -format single
```

Backup Task: `-backup <files>..`

Use the `-backup` task flag to execute a backup creating a new backup set.

- Alternative formats: `-b <files>..`

- Use one or more `<files>` parameters to specify the source files and/or directories to back up.

The standard wildcard symbols (`*` and `?`) can be used to specify file name templates.

Wildcard symbol	Matches
`*` (asterisk)	Zero or more characters in that position.
`?` (question mark)	Exactly one character in that position.

All files specified after a `-backup` task flag will be recorded in one new backup set.

Use the <u>-append mode flag</u> to choose whether to append the new backup set to the existing ones or to replace some or all of the previously recorded backup sets. The `-append` mode flag is set to "on" by default, that is, if `-append` on/off flag has not been specified, the new backup set will be appended to the previous ones existing on the tape.

CAUTION: If `-append` mode has been set to "off", the new backup set will replace all content previously recorded on the tape!

Optionally, include the <u>-subdirectories on/off mode flag</u> to recurse through subdirectories.

Optionally, supply the <u>-crc data integrity verification flag</u> to specify a hash data verification algorithm (CRC) for the new backup set.

To perform an <u>incremental backup</u>, that is, to back up only the files that have changed or been added anew since the last backup set was recorded, set the <u>-incremental on/off mode flag</u> to "on". The newly created backup set will be marked as incremental. If no changed or added files have been detected, `tapecon` will create no backup sets.

Remember to provide all backup mode specification flags **before** the `-backup` task flag itself.

If during backup the tape media volume will become filled up, `tapecon` will prompt to insert a new media volume to continue the backup onto it, thus creating a **multi-volume backup**.

CAUTION: All the content on the subsequently supplied volumes of a multi-volume backup will be replaced, even if the `-append` mode has been set to "on"!

Examples:

Backup all files from `C:\Data` directory, without its subdirectories, replacing all the previous content on the tape:

```
tapecon -d:0 -append:OFF -backup C:\Data
```

Backup only the files from `C:\Data` directory that have changed since the most recent backup set:

```
tapecon -d:0 -inc -backup C:\Data
```

Backup all .DOC* (such as .DOC and .DOCX) and all .TXT files from the directory `C:\My Documents` including its subdirectories, appending the new backup set to the backup sets already on the tape:

```
tapecon -d:0 -subdir -backup "C:\My Documents\*.doc*" "C:\My Documents\*.txt"
```

Restore Task: `-restore [<set_index>] [<files>..]`

Use the `-restore` task flag to restore previously backed up files.

- Alternative formats: `-r [<set_index>] [<files>..]`
- Optionally supply `<set_index>` parameter to restore the files from a particular set. If no `<set_index>` has been specified, the files from the last (most recently recorded) backup set will be restored. S. Backup set indexes on how to specify a backup set index.

`<set_index>` value	Backup set selected
None specified	The last (most recently recorded)
Positive number from `1` to `<number_of_sets>`	From the oldest (`1`) to the most recent (`<number_of_sets>`) set
`0`	The last (most recently recorded)
Negative number from `-1` to `-(<number_of_sets> - 1)`	From the second most recent (`-1`) to the oldest (`-(<number_of_sets> - 1)`)

Here `<number_of_sets>` denotes the number of the backup sets recorded on the tape media.

- Optionally use `<files>` parameter to specify the files to be restored.

The standard wildcard symbols (`*` and `?`) can be used to specify file name templates. S. `-backup` task flag for the wildcard symbols. If no `<files>` parameter has been specified, all files from the backup set will be restored.

By default, the files will be restored to their original locations. To specify an alternative location, include `-target` mode flag before the `-restore` task flag. If the directory name is omitted, the target directory setting is cleared, that is, the files will be restored to their original location.

Optionally use the `-subdirectories` one/off mode flag to restore the directory structure for the files being restored. If an alternative target directory has been specified using `-target` mode flag, the directory structure will be restored under the target directory.

Optionally use the `-existing` mode flag to specify how to handle the restored files if a file with the same name already exists at the target location. If the flag has not been specified, `tapecon` will keep both files by renaming restored files.

When restoring files from an incremental backup set, `tapecon` will automatically refer to all the previous incremental backup sets to restore the most up-to-date version of each file. To disable this behavior and restore all files from the specified backup set only, set the `-incremental` on/off mode flag to "off".

When restoring files from a backup set that is a part of a **multi-volume backup**, `tapecon` will prompt to insert the previous tape volume(s) to complete restoring. Since every volume of a multi-volume backup keeps the information only of its predecessor volumes, start restoring a multi-volume backup from the last tape volume.

Remember to provide all restore mode specification flags **before** the `-restore` task flag itself.

Examples:

Restore all files from the last (most recent) backup set to their original locations, renaming restored files if a file under the same name already exists:

```
tapecon -d:0 -restore
```

Restore all .DOC* (such as .DOC and .DOCX) and all .TXT files from the second most recent backup set (set index -1) to the directory `C:\My Restored Documents`, recreating their original directory structure underneath and skipping the files that already exist at the target location:

```
tapecon -d:0 -target "C:\My Restored Documents" -subdir –existing:skip -restore -1
*.doc* *.txt
```

Verification Task: `-verify [<set_index>] [<files>..]`

Use the `-verify` task flag to verify backed up files by comparing them to the original ones.

- Alternative formats: `-y [<set_number>] [<files>..]`
- The optional parameters `<set_number>` and `<files>` have the same form and meaning as for the `-restore` task flag. S. Backup set indexes on how to specify a backup set index.

Verification compares the backed-up files from the specified backup set on tape with the original ones. The result is reported for every file.

To perform the validation of the backed-up files using only the stored data integrity information without comparing them with the original ones, use the validation task.

The verification task may be performed right after the backup task to verify the files in the newly recorded backup set.

Examples:

Verify all files from the last (most recent) backup set by comparing them to the original ones:

```
tapecon -d:0 -verify
```

Backup all files from the directory `C:\My Data` including subdirectories, appending the new backup set to the backup sets already on the tape, then verify the newly backed up files by comparing to the original ones:

```
tapecon -d:0 -subdir -backup "C:\My Data" -verify
```

Verify all .DOC* (such as .DOC and .DOCX) and all .TXT files from the second most recent backup set (set index -1) by comparing to the original ones:

```
tapecon -d:0 -y -1 *.doc* *.txt
```

Validation Task: `-validate [<set_index>] [<files>..]`

Use the `-validate` task flag to validate backed up files by using the stored data integrity information.

- Alternative formats: `-v [<set_number>] [<files>..]`
- The optional parameters `<set_number>` and `<files>` have the same form and meaning as for the `-restore` task flag. S. Backup set indexes on how to specify a backup set index.

The backed-up files are validated by the hash (CRC) algorithm using the stored data integrity information. No comparison with the original files is performed. The result is reported for every file.

To perform verification comparing the backed-up files on tape with the original ones, use the verification task.

The validation task may be performed right after the backup task to validate the files in the newly recorded backup set.

Examples:

Validate all files from the last (most recent) backup set by the hash (CRC) algorithm using the stored data integrity verification information:

```
tapecon -d:0 -validate
```

Backup all files from the directory `C:\My Data` including subdirectories, appending the new backup set to the backup sets already on the tape, then validate the newly backed up files:

```
tapecon -d:0 -subdir -backup "C:\My Data" -validate
```

Validate all .DOC* (such as .DOC and .DOCX) and all .TXT files from the second most recent backup set (set index -1):

```
tapecon -d:0 -v -1 *.doc* *.txt
```

List Task: `-list [<set_index_from> [<set_index_to>]] [<files>..]`

Use the `-list` task flag to list the content of backup sets on the tape.

- Alternative formats: `-l [<set_index_from> [<set_index_to>]] [<files>..]`
- The optional parameters `<set_index_from>`, `<set_index_to>` and `<files>` have the same form and meaning as for the `-restore` task flag. S. Backup set indexes on how to specify a backup set index.

For the `-list` task flag, either a particular backup set or a range of backup sets can be specified. If no backup sets are specified, the content of all the backup sets on the tape media will be listed.

Optionally use the `-subdirectories` mode flag to list the full file names including the full directory path part. Otherwise, only file names (without the directory part) will be listed.

By default, the `-list` task flag will structure incremental backup sets to list only the most up-to-date files from each of the incremental sets. To disable this behavior and list the full content of the incremental backup sets, set the optional `-incremental` mode flag to off.

Examples:

List the content of all backup sets on the tape media in drive 0:

```
tapecon -d:0 -list
```

List the content of all backup sets on the tape media in drive 0 without incremental backup set structuring:

```
tapecon -d:0 -inc- -list
```

List all .DOC* (such as .DOC and .DOCX) and all .TXT files from the three most recent backup sets (set indexes from -2 to 0) including the full directory path information for each file:

```
tapecon -d:0 -subdir -list -2 0 *.doc* *.txt
```

Eject Task: -eject

Use -eject task flag to unload (eject) the media from the drive.

- Alternative formats: -j

Provide the -eject task flag as the last flag on the tapecon command line to unload (eject) the media from the drive upon the completion of the previous tasks.

Examples:

Backup all files from the directory C:\My Data including subdirectories, appending the new backup set to the backup sets already on the tape; verify the newly backed up files by comparing to the original ones; finally, eject the tape:

```
tapecon -d:0 -subdir -backup "C:\My Data" -verify -eject
```

Help Task: -help [<flag_without_dash>..]

Use -help task flag to display tapecon usage instructions.

- Alternative formats: -h or -?
- Parameter <flag_without_dash>: Optionally specify one or more flags (without leading dash) to display help for. Specify index to display explanation of backup set indexing. To display general tapecon usage instructions, specify no parameter.

Invoking tapecon without any flags will display the general tapecon usage instructions.

Examples:

Display help on -drive flag:

```
tapecon -help drive
```

Display help on -backup and -restore flags (notice the flags specified in their short form):

```
tapecon -? b r
```

Display explanation of backup set indexing:

```
tapecon -help index
```

Display general `tapecon` usage instructions:

```
tapecon -help
```

or

```
tapecon
```

Mode Flags

The mode flags specify how to fulfill the tasks subsequent on the `tapecon` command line. For example, the <u>-subdirectories mode flag</u> directs `tapecon` to recurse subdirectories when executing every subsequent <u>-backup task</u>.

Every mode flag applies to all the subsequent task flags. Exception: the <u>-append mode flag</u> applies only to the single subsequent -backup task flag, after which the -append mode is reset to its default value "on".

Quiet Mode: `-quiet [on|off]`

Use the `-quiet` on/off mode flag to enable or to suppress `tapecon` interactive prompts.

- Alternative formats: `-q [on|off]`
- Used for task flags: all.

Parameter value	Behavior
on	Disable interactive prompts – "quiet" mode on. Assume positive responses.
off	Enable interactive prompts – "quiet" mode off. Await user input.
None specified	Disable interactive prompts – "quiet" mode on. Assume positive responses.

- Default setting (when the flag is not provided): "off"

If the `-quiet` mode flag has been set to "off", `tapecon` will not await user input to the interactive prompts, much rather assume positive responses ("Yes") to all interactive prompts.

The "off" setting might be convenient for batch operations, when `tapecon` is invoked from an automated script.

CAUTION: If the `-quiet` mode is set to off, the caution confirmation prompts before overwriting any existing data will be automatically confirmed and so the data overwritten!

Examples:

Backup all files from `C:\Data` directory, without its subdirectories, replacing all the previous content on the tape – without asking for confirmation to overwrite any existing data on the tape:

```
tapecon -d:0 -append- -quiet- -backup C:\Data
```

Preferred Block Size: `-blocksize [<size>]`

Use the `-blocksize` mode flag to specify the preferred block size to use for recording subsequent backup sets.

- Alternative formats: `-block [<size>]` , `-z [<size>]`
- Used for task flags: `-backup`

Parameter value	Behavior
Integer value < 1,024	Set the preferred block size to the specified value in kilobytes (KB)
Integer value > 1,024	Set the preferred block size to the specified value in bytes
`0`	Set the block size to the tape drive's default value
None specified	Set the preferred block size to the default `tapecon` value 16 KB

- Default setting (when the flag is not provided): use the default `tapecon` value 16 KB.

The actual block size will be set to the supported value closest to the specified preferred value. Most drives support block sizes between 1 KB and 64 KB. `tapecon` does not support setting block sizes below 1 KB.

Larger block sizes may accelerate the recording and restoring backup sets and are recommended for backing up larger files. Smaller block sizes help use the tape space more sparingly and are recommended for backing up numerous smaller files.

The setting applies to all the backup sets created by the subsequent `-backup` tasks on the command line. Various backup sets on the same tape media may be recorded using different block sizes.

Examples:

Backup all files from `C:\Data.SMALL` directory, with its subdirectories, using block size 4 KB; then backup all files from `C:\Data.MEDIUM` directory, with its subdirectories, using the default `tapecon` block size 16 KB; finally, backup all files from `C:\Data.LARGE` directory, with its subdirectories, using block size 64 KB:

```
tapecon -d:0 -s -z 4 -b C:\Data.SMALL -z -b C:\Data.MEDIUM -z 64 -b C:\Data.LARGE
```

Tape Filemark Usage Mode: `-filemarks [on|off]`

Use the `-filemarks` on/off mode flag to enable or disable usage of tape filemarks in the subsequently recorded backup sets.

- Alternative formats: `-fm [on|off]`
- Used for task flags: `-backup`

Parameter value	Behavior
on	Use filemarks in the subsequent backup tasks.
off	Do not use filemarks in the subsequent backup tasks.
None specified	Use filemarks in the subsequent backup tasks.

- Default setting (when the flag is not provided): "off".

If the `-filemarks` on/off mode flag has been set to "on", `tapecon` will insert tape filemarks between files in the backup sets recorded by the subsequent backup task flags.

Tape filemark usage can accelerate accessing separate files when restoring them from a backup set – at the price of slowing down the backup recording process. It is therefore recommended to leave this mode at the default "off" setting, unless backing up a few large files, such as long video clip files or archive files like .ZIP or .TAR.

On some drives, such as DST VS1, `tapecon` does not support tape filemark usage. In such cases the tape filemark usage is always set to "off".

The setting applies to all the backup sets created by the subsequent -backup tasks on the command line. Various backup sets on the same tape media may be recorded using different data tape filemark usage modes.

Subdirectories Mode: `-subdirectories [on|off]`

Use the `-subdirectories` on/off mode flag to specify how to use subdirectory information during backup or restore operations.

- Alternative formats: `-subfolders [on|off]`, `-subdir [on|off]`, `-s [on|off]`
- Used for task flags: -backup , -restore , -list

Parameter value	Behavior
on	Use subdirectory information
off	Do not use subdirectory information
None specified	Use subdirectory information

- Default setting (when the flag is not provided): "off".

Task	Behavior when **-subdirectories** is set to "on"	Behavior when set to "off" (default)
-backup	Recurse subdirectories for source files	Do not recurse subdirectories
-restore	Restore the directory structure for restored files	Do not restore the directory structure
-list	Output the full directory path name of each file listed	Output just the file names without the directory part

Examples:

Backup all .DOC* (such as .DOC and .DOCX) and all .TXT files from the directory `C:\My Documents` including subdirectories, appending the new backup set to the backups already on the tape:

```
tapecon -d:0 -subdir -backup "C:\My Documents\*.doc*" "C:\My Documents\*.txt"
```

Restore all .DOC* (such as .DOC and .DOCX) and all .TXT files from the second most recent backup set (set index -1) to the directory `C:\My Restored Documents`, restoring their original directory structure:

```
tapecon -d:0 -target "C:\My Restored Documents" -subdir -restore -1 *.doc* *.txt
```

Data Integrity Verification (CRC) Mode: `-crc [on|off|<algorithm>]`

Use the `-crc` mode flag to specify the data integrity verification (CRC) algorithm to use during backup operations.

- Alternative formats: `-c [on|off|<algorithm>]`
- Used for task flags: `-backup`
- Possible `<algorithm>` parameter values:

`<algorithm>` value	Data integrity algorithm to use
`none`	none
`off`	none
None specified	Crc32
`on`	Crc32
`crc32`	Crc32
`32`	Crc32
`crc64`	Crc64
`64`	Crc64
`xxhash32`	Xxhash32
`xx32`	Xxhash32
`xxhash3`	Xxhash3
`xx3`	Xxhash3
`xxhash64`	Xxhash64
`xx64`	Xxhash64
`xxhash128`	Xxhash128
`xx128`	Xxhash128
`128`	Xxhash128

- Default setting (when the flag is not provided): use Crc32 algorithm.

The `-crc` mode flag can be specified either as an on/off flag or as a parameter flag. As an on/off flag it specifies either using Crc32 data integrity verification algorithm ("on" setting) or disabling data verification ("off" setting).

Providing the `<algorithm>` parameter allows to specify other algorithms to use instead of Crc32.

Using a data integrity verification algorithm helps ensure the integrity of the data stored on the tape.

The setting applies to all the backup sets created by the subsequent `-backup` tasks on the command line. Various backup sets on the same tape media may be recorded using different data integrity algorithms.

Examples:

Backup all files from `C:\Data` directory, without its subdirectories, using Crc32 data integrity algorithm; then backup all files from `C:\Important Data` directory, without its subdirectories, using Crc64 data integrity algorithm:

```
tapecon -d:0 -crc -backup C:\Data -crc 64 -backup "C:\Important Data"
```

Append Mode: `-append [on|off|<after_set_index>]`

Use the `-append` mode flag to specify an existing backup set to append the new backup set after.

- Alternative formats: `-a [on|off|<after_set_index>]`
- Used for task flags: `-backup`
- The optional parameter `<after_set_index>` has the same form as for the `-restore task flag`. S. Backup set indexes on how to specify a backup set index.

Parameter value	Behavior
`on`	Append the new backup set after all the latest backup set
`off`	Overwrite any existing content on the tape media with the new backup set
`<after_set_index>`	Add the new backup set after the backup set with index `<after_set_index>`
None specified	Append the new backup set after all the latest backup set

- Default setting (when the flag is not provided): "on".

The `-append` mode flag can be specified either as an on/off flag or as a parameter flag. As an on/off flag it specifies whether to add the new backup set after all the existing ones on the tape media.

CAUTION: If `-append` mode has been set to "off", the new backup set will replace all content previously recorded on the tape!

Providing the `<after_set_index>` parameter allows to specify an existing backup set after which the new backup set will be added.

CAUTION: All the existing backup sets following the specified one will be overwritten!

If the `-incremental mode` has been set to "on", then the `-append` mode is also assumed "on". In this case the `<after_set_index>` parameter, if specified, is ignored.

The `-append` mode setting applies only to the subsequent `-backup` task on the command line. After executing this `-backup` task, the `-append` mode setting will be reset to its default value "on". To specify a different `-append` mode setting to each further `-backup` task, provide a dedicated `-append` mode flag before the `-backup` task flag.

Examples:

Backup all files from `C:\Data` directory, without its subdirectories, replacing all the previous content on the tape:

```
tapecon -d:0 -append:OFF -backup C:\Data
```

Backup all files from `C:\Data` directory, without its subdirectories, adding the new backup set to the previous content on the tape:

```
tapecon -d:0 -append -backup C:\Data
```

Backup all files from `C:\Data` directory, without its subdirectories, replacing the last two backup sets on the tape (the 3rd most recent backup set, under the index -2, will be kept; s. the illustration to Backup set indexes):

```
tapecon -d:0 -append -2 -backup C:\Data
```

Incremental Mode: `-incremental [on|off]`

Use the `-incremental` on/off mode flag to specify whether to conduct an incremental backup and how to use incremental backup information.

- Alternative formats: `-inc [on|off]`, `-i [on|off]`
- Used for task flags: `-backup`, `-restore`, `-list`

Parameter value	Behavior
on	Conduct incremental backup; use incremental backup set information
off	Conduct regular (non-incremental) backup; do not use incremental backup set information
None specified	Conduct incremental backup; use incremental backup set information

- Default setting (when the flag is not provided): Conduct regular (non-incremental) backup; yet do use incremental backup set information.

Task	Behavior when `-incremental` is set to "on"	Behavior when set to "off"	Default behavior (when `-incremental` not provided)
-backup	Conduct incremental backup	Conduct regular (non-incremental) backup	Conduct regular (non-incremental) backup
-restore	Restore files using incremental backup information	Restore just from the specified backup set, not using incremental backup information	Restore files using incremental backup information
-list	List files using incremental backup set information	List all files not using incremental backup information	List files using incremental backup set information

Examples:

From the directory `C:\My Data`, backup the files that have changed or been added since the last backup, appending the new backup set to the backup sets already on the tape:

```
tapecon -d:0 -inc -backup "C:\My Data"
```

Restore the most recent backup set to the directory `D:\Restored Files`. If the backup set is incremental, restore the most up-to-date version of each file across all the previous backup sets:

```
tapecon -d:0 -target "D:\Restored Files" -restore
```

Restore the second most recent backup set to the directory `D:\Restored Files`. Restore all the files as contained in the backup set, even if it is an incremental one:

```
tapecon -d:0 -target "D:\Restored Files" -incremental:OFF -restore -1
```

List the content of the tape media first using the incremental backup set information (list only the most up-to-date files from the incremental backup sets), then without using incremental backup set information (list all the files in every backup set regardless their up-to-date status):

```
tapecon -d:0 -list -inc- -list
```

Target Mode: -target [<directory>]

Use the `-target` mode flag to specify a directory to restore files to.

- Alternative formats: `-t [<directory>]`
- Used for task flags: -restore.

Parameter value	Behavior
`<directory>`	Restore files to the directory `<directory>`.
None specified	Restore files to their original locations.

- Default setting (when the flag is not provided): Restore files to their original locations.

Examples:

Restore all files from the last (most recent) backup set to their original locations, renaming restored files if a file under the same name already exists:

```
tapecon -d:0 -restore
```

Restore all files from the last (most recent) backup set to the directory `D:\Restored Files`, recreating their original directory structure underneath:

```
tapecon -d:0 -t "D:\Restored Files" -s -r
```

Existing Mode: -existing [<how_to_handle>]

Use the `-existing` mode flag to specify how to handle the restored files if a file with the same name already exists at the target location.

- Alternative formats: `-e [<how_to_handle>]`
- Used for task flags: -restore.

`<how_to_handle>` value	Alternative forms	Behavior on restoring a file with the same name as an existing file
`overwrite`	`o`	Overwrite the existing file with the restored file
`skip`	`s`	Skip the file being restored
`keepboth`	`keep`, `both`, `k`, `rename`	Keep both files (rename the restored file)
None specified		Keep both files (rename the restored file)

- Default setting (when the flag is not provided): keep both files by renaming restored files.

CAUTION: If `-existing` mode is set to `overwrite`, exiting files will be overwritten without confirmation!

Examples:

Restore all files from the last (most recent) backup set to their original locations, renaming restored files if a file under the same name already exists:

```
tapecon -d:0 -restore
```

The same as above:

```
tapecon -d:0 -existing:keep -restore
```

The same as above, but skipping existing files:

```
tapecon -d:0 -existing:skip -restore
```

Restore all files from the last (most recent) backup set to the directory `D:\Restored Files`, recreating their original directory structure underneath and overwriting existing files:

```
tapecon -d:0 -t "D:\Restored Files" -s -e:o -r
```