

# Understanding File Attributes

Author: Conrad Chung, [2BrightSparks](#)

We use digital files every day, but it's relatively rare we need to recognize the attributes of a digital file. This article is all about understanding File Attributes.

You might for example have wondered how the Windows operating system differentiates files that are visible from those that are hidden. Or let's say a text editor program pops up an error message informing you the file is "read-only" when you're trying to save changes to the file - you might ask yourself: how does the text editor know that the file is read-only? File Attributes tag files with additional information that Windows uses to act on.

Knowing a little about File Attributes will help you understand why certain things happen with files, and why certain things can't happen if an attribute is defined in a certain way.



## What are File Attributes and What are They Used For?

File attributes are pieces of information associated with every file and directory that includes additional data about the file itself or its contents. They can exist in only one of two states – Set or Cleared; similar to an On or Off state. Attributes can be in files, directories, volumes and certain system objects. They are used by the operating system and software applications to define file system behavior.

## How are File Attributes Used?

File attributes are pieces of information associated with every file and directory that includes additional data about the file itself or its contents. They can exist in only one of two states – Set or Cleared; similar to an On or Off state. Attributes can be in files, directories, volumes and certain system objects. They are used by the operating system and software applications to define file system behavior.

## Listing of the Commonly Used Attributes

A byte stores the attributes of a file, with each specific attribute assigned to a bit of a byte. To enable a certain attribute, the system will assign a 'one' to the corresponding bit, which represents the 'On' state. (These are referred to as flagging or setting the attribute). A Windows operating system (Win32) stores the file attributes as 32-bit quantity, while the original MS-DOS file attributes only have 8 bits to store file attributes.

Below are the common attributes and the bits that represent them:

Attribute	Bit Value
Read-Only (R)	00000001
Hidden (H)	00000010
System (S)	00000100
Directory (D)	00010000
Archive (A)	00100000
Not Content Indexed (I)	10000000000000

A file may have more than one attribute mapped to it by adding the bits to form the byte attribute. A read-only, hidden directory would have the attribute byte of 00010011, a result of the three attributes added together. Below are the descriptions of each attribute:

- **Read-Only (R)** : Read-Only attribute will prevent software programs from saving changes to a file. This is useful if you want to write protect a file. Note that a Read-Only file will not prevent it from being deleted.
- **Hidden (H)** : A file marked with the hidden attribute will be hidden from view under normal viewing conditions.
- **System (S)** : A file or directory used exclusively by the operating system which should not be altered or deleted.
- **Directory (D)** : This attribute is tagged to folders or sub-folders to differentiate them from files.
- **Archive (A)** : This bit is used by software applications that modify files as well as backup software as a “communication link”. Some backup software (for example SyncBackPro and SyncBackSE) allows incremental backups by the user, which only backs up files that have changed since the previous backup. When the backup software archives or backs up the file, it resets the archive bit (tagging it *zero* or *Off*). Any software that subsequently make changes to the file is expected to set the archive bit. Thus when the backup software runs the next time, it will be able to identify the modified files by analyzing the archive bits, and backs up those files with the modified bits.

Note that certain programs may modify the files without marking the archive attribute. If the backup software uses incremental backups to backup these files, it will rely on the software to set the bit appropriately. It is therefore important to note that you should not rely solely on this setting to ensure critical files are backed up.

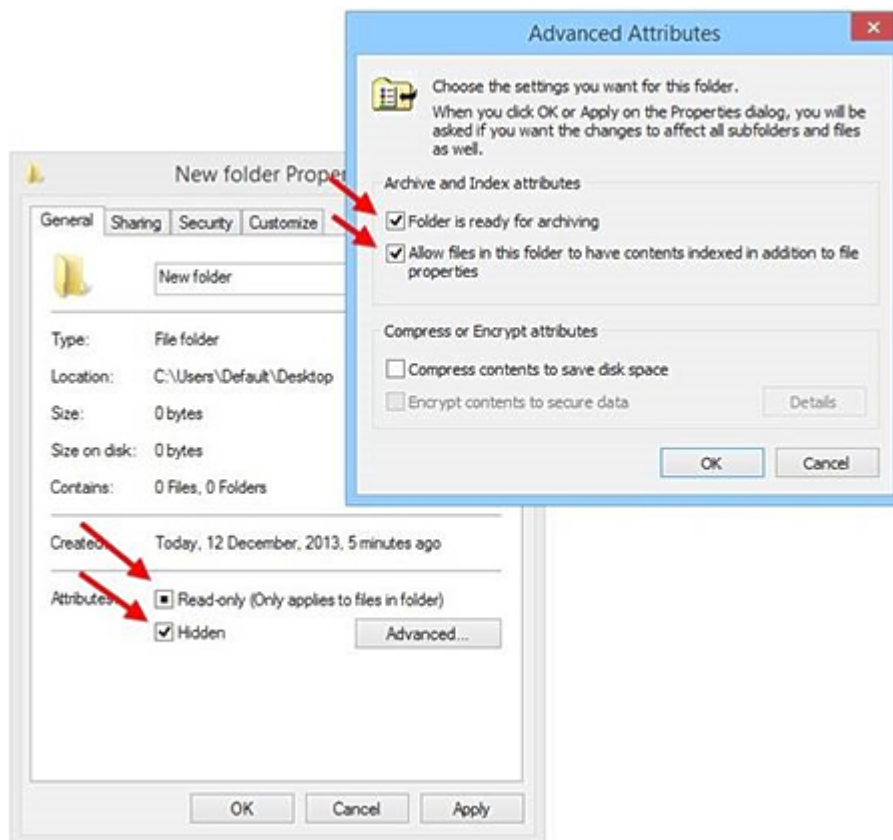
**Not Content Indexed (I)** : Windows has a search function which indexes all the files and directories in the drive in order to achieve faster results. By default, this attribute is not set, thereby allowing the operating system to include the file into the indexing list. A user could choose to opt out certain files by setting this attribute to *On*. When set, Windows will skip files that have this option checked.

## How to View and Modify attributes?

There are a few ways to view and change attributes. Two of them are via the file or folder properties from Windows Explorer or using the ATTRIB command from MS-DOS prompt (type CMD command to bring up the DOS window).

## Viewing Attributes from Windows Explorer

Right-clicking and selecting Properties on any file or folder in Windows Explorer will bring up the Properties window, which shows the attributes of the selected item.



**Note:** The extended attributes (compress and encrypt) will not be shown when you use the **ATTRIB** command. However you can view these attributes by executing the **Compact** and **Cipher** commands, respectively.

## Viewing Attributes from DOS prompt

To view attributes, you can type ATTRIB command from the DOS prompt.

Typing **ATTRIB /?** will bring up a list of all the syntax available to query a file.

```
C:\Windows\system32>attrib /?
Displays or changes file attributes.

ATTRIB [+R | -R] [+A | -A ] [+S | -S] [+H | -H] [+I | -I]
      [drive:][path][filename] [/S [/D] [/L]]

+     Sets an attribute.
-     Clears an attribute.
R     Read-only file attribute.
A     Archive file attribute.
S     System file attribute.
H     Hidden file attribute.
I     Not content indexed file attribute.
[drive:][path][filename]
      Specifies a file or files for attrib to process.
/S    Processes matching files in the current folder
      and all subfolders.
/D    Processes folders as well.
/L    Work on the attributes of the Symbolic Link versus
      the target of the Symbolic Link
```

To view what attributes are assigned to a file, type the following:

```
ATTRIB C:\Folder\Filename.ext
```

From here, we can see that the TEST.TXT file has the Archive, Hidden and Read-Only attribute enabled (A, H & R are shown).

```
C:\>ATTRIB C:\FOLDER\TEST.TXT
A HR      C:\FOLDER\Test.txt
C:\>_
```

Using the ATTRIB command allows a user to *Set* or *Clear* an attribute of a file. For the above example, if we wish to remove the attribute of Hide and Read-Only from the file, we could type:

```
ATTRIB -H -R C:\FOLDER\TEST.TXT
```

```
C:\>ATTRIB -H -R C:\FOLDER\TEST.TXT
C:\>
C:\>ATTRIB C:\FOLDER\TEST.TXT
A      C:\FOLDER\Test.txt
C:\>
C:\>
```

The example above shows that the hidden and Read-Only attribute have been removed from TEST.TXT.

## Conclusion

This article has hopefully helped you understand a little more about Windows file attributes. Being familiar with file attributes will help you better understand how the file system works in a Windows environment. You could use this knowledge to your advantage by selecting the Fast Backup (scanning the Archive Attribute) option in [SyncBackSE](#) and [SyncBackPro](#) to greatly reduce the time needed for each backup, thereby increasing your productivity and freeing up your system resources for other tasks.