

SourceSafe Q & A: Read-Only Files

SourceSafe Question

Some of the files in my local directory have been tagged so that the operating system will not let me change or delete them. When and why does SourceSafe do this?

The Basic Answer

SourceSafe works, in general, by the following rules.

- If you have a file checked out, your local copy is writable (*ie* the file can be modified).
- If you do not have a file checked out, your local copy is not writable (cannot be modified).

The exact method of making a file "not writable" varies with the operating system: a DOS file is made *read-only*, a Macintosh file is *locked*, and a UNIX file has *write permission turned off for all users*. But the basic idea is always the same: when you do not have a file checked out, SourceSafe tags the file unmodifiable. Hence, SourceSafe commands such as Add, Update, Uncheckout, and Get leave your file read-only; Checkout makes it writable.

Why SourceSafe Makes Files Read-Only

The first important point to understand here is that *making files read-only is not for security purposes*. SourceSafe is not trying to prevent you forcefully from modifying the file. If you Get a file, and then manually make it writable and start making changes, you have not endangered SourceSafe's Checkout system. (Remember, the true "master" copy of every file is in SourceSafe; and SourceSafe will not let you Update a file that you do not personally have checked out, regardless of its read-only status!)

That being said, there are two principal reasons SourceSafe locks your local copy of a file.

1. The read-only attribute serves as a *reminder to the user* that the file is not checked out. Most editors will alert you when you try to modify a read-only file, thus reminding you to Checkout the file before modifying it. This can save you the great agony of spending hours modifying a file, only to discover that you forgot to check it out!
2. The read-only attribute serves as a *signal to SourceSafe* of whether you have been modifying a file. When a file is read-only, SourceSafe assumes that the file is safely in SourceSafe; hence, when you attempt to Get the file, SourceSafe will cheerfully overwrite your local copy. But a writable file may be one you have checked out, or which you have set writable and modified, and may therefore contain information which exists nowhere else; by default, SourceSafe will never overwrite such a file.

Changing SourceSafe's Use of Read-Only Files

The behavior described above is SourceSafe's *default behavior*; however, like most assumptions made by SourceSafe, this can be changed if your needs are different. There are two ways to modify SourceSafe's use of read-only files.

The first is through the SS.INI variable `Use_Readonly`. If the following line is placed in your SS.INI file, SourceSafe will never tag files read-only.

```
Use_Readonly = No
```

This line signals SourceSafe never to make a file read-only, and not to use the read-only attribute as its signal that a file should not be overwritten. (SourceSafe will use the file's actual Checkout status in this case.) As with all INI variables, this can be set per-user in SS.INI or globally in SYSTEM.INI; and it can be set project-by-project or generally. See the description of SS.INI in the SourceSafe User's Manual or On-Line Help for more details.

The other option, available from the SourceSafe command-line only, is to change SourceSafe's use of the read-only flag for an individual command. This is done via the **-W** (*writable*) parameter. If you use the **-W** with a command such as Get, Add, or Update, which would normally tag a file read-only, the file will be made writable. If you have `Use_Readonly` set to No by default, you can use the **-W-** parameter to override it for an individual command and make the file read-only.