

SourceSafe Q & A: The File Modification Date

SourceSafe Question

I'm confused by the modification date/time that SourceSafe gives my files when it retrieves them to a local directory. How does it work, and why?

What Modification Date/Time SourceSafe Gives Your Files

When you Get a file (or Checkout a file, or in any other way copy a file from SourceSafe's database to your local directory), SourceSafe uses the following rules to determine the date/time it gives your file.

- If you *already have* a local copy of the file; *and* if that file is identical to the file you are getting; SourceSafe does nothing. In that case, the file's timestamp remains unchanged.
- If SourceSafe does have to get the file, it gives it the *current* date/time; that is, when the Get is performed.

The latter case may occur if you do not have the file in your current directory; or, if your copy or the file is different from what you are getting. The key point here is that whenever SourceSafe actually has to copy the file to your directory, the file's date of last modification is set to *right now*. This may not be what you would expect at first, and it is worth saying a few words about why the system works this way.

Why SourceSafe Works This Way

The modification date/time is used by many compilers and make systems as a signal of *whether a file has changed, and therefore needs to be remade*. SourceSafe's use of the date/time is designed to optimize this process, ensuring that you will compile when you need to, and not when you don't.

Consider the following example.* A C programmer has the file HELP.C in SourceSafe, and in his local directory. His compiler will turn this code file into the object module HELP.OBJ. At any given time during the development cycle, both files--HELP.C and HELP.OBJ--probably exist in the same directory. Each time the developer types MAKE, his Make utility compares the date/time stamp of HELP.C with HELP.OBJ to determine if it should compile, using the following rules.

1. If HELP.C is more recent than HELP.OBJ, the code has changed, and a recompile is necessary.
2. If HELP.OBJ is more recent than HELP.C, the code has already been compiled, and no recompile is necessary.

* The example is given based on the C language; however, the logic described in the example applies to many different languages and development systems.

This behavior is simple, logical, and relatively universal; it ensures that every Make builds the entire program, from the most recent version of the code, as quickly as possible.

Now consider the ramifications for SourceSafe. You compile HELP.C into HELP.OBJ, giving the latter a new date/time. If you Make again without changing anything, HELP.C will not be recompiled. Satisfied with your changes, you Update HELP.C into SourceSafe; and the next morning, you Get HELP.C again.

If HELP.C has not changed, SourceSafe recognizes that its own copy and yours are identical. It does not do a Get, the file date/time does not change, and the file is not recompiled--which is as it should be.

If another developer has changed HELP.C, SourceSafe recognizes that the file is different. It does get HELP.C, and makes its date/time *now*. When you make, the file is recompiled--which, again, is as it should be. Hence, whether someone has changed the file or not, SourceSafe's behavior sends the correct signal to the Make utility.

To take one final example: suppose that you now Get an *old* version of HELP.C. If SourceSafe did what you might expect, and gave the .C file its old date/time, the file would not recompile, since the .C file would be older than the .OBJ. In this case, you might wonder why your program was not doing what it used to--when in fact, the reason is that you are still using an object module from a previous compilation! However, this is not what happens. Instead, SourceSafe recognizes that what you are getting (the old version) is different from what you have (the most recent), and gets the file, giving it the current date/time. This forces a recompile, thus ensuring that the program you build is in line with the code you got.

Changing SourceSafe's Use of the Date/Time Stamp

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 the file modification time.

The first is through the SS.INI variable `Restore_ModTime`. If the following line is placed in your SS.INI file, SourceSafe will always give files the date/time of their actual last modification, instead of the current date/time.

```
Restore_ModTime = Yes
```

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 file timestamp for an individual command. This is done via the **-GM** parameter. If you use the **-GM** parameter with a command such as Get or Checkout, the file will be given the timestamp of its actual last modification. If you have `Restore_Modtime` set to Yes by default, you can use the **-GM-** parameter to override it for an individual command and give the files the current date/time.