

---

Subject: Re: Compile or not compile?

Posted by [JD Smith](#) on Mon, 21 Mar 2005 16:40:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Fri, 04 Mar 2005 03:42:50 -0800, m\_schellens@hotmail.com wrote:

- > 2. Files which have been added to IDL's path after it started. IDL
- > scans its path at startup, and only files which exist at that time
- > can be found automatically.
- >
- > Under linux at least this is not true. Here IDL always searches through
- > the path again. Maybe it is different for the library path (the initial
- > !PATH setting), I haven't checked it there.
- > And what happens if !PATH is changed?

I suspect you are referring IDL<6.0, which was when path caching was added to IDL. I think it must depend on whether your IDL is caching its path, which it is by default starting at IDL 6.0. For me (under Linux), newly created subdirectories and routines are not discovered:

The PATH\_CACHE procedure is used to control IDL's use of the path cache. By default, as IDL searches directories included in the !PATH system variable for .pro or .sav files to compile, it creates an in-memory list of all .pro and .sav files contained in each directory. When IDL later searches for a .pro or .sav file, before attempting to open the file in a given directory, IDL checks the path cache to determine whether the directory has already been cached. If the directory is included in the cache, IDL uses the cached information to determine whether the file will be found in that directory, and will only attempt to open the file there if the cache tells it that the file exists. By eliminating unnecessary attempts to open files, the path cache speeds the path searching process.

The path cache is enabled by default, and in almost all cases its operation is transparent to the IDL user, save for the boost in path searching speed it provides. Because the cache automatically adjusts to changes made to IDL's path, use of PATH\_CACHE should not be necessary in typical IDL operation. It is provided to allow complete control over the details of how and when the caching operation is performed.

The other thing to note is that the PATH is not cached on startup, only after the first time it is read. So try something like:

```
IDL> .run file_in_my_path
```

create another\_file\_in\_my\_path.pro in the same location as  
file\_in\_my\_path.pro

IDL> .run another\_file\_in\_my\_path.pro

This will fail if the paths are cached.

JD

---