

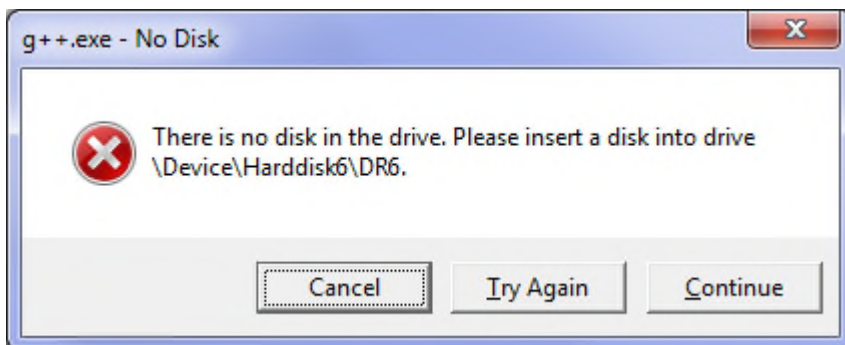
## MinGW Problem with Drives that Don't Exist

### Jon Schnute - December, 2013

**Problem:** I assume that you have a Windows computer with [MinGW](#) and the Gnu C++ compiler installed. If the MinGW\bin directory is on the path, you should be able to open a command window and find the version number of g++ with a command session like this:

```
C:\>g++ --version
g++ (GCC) 4.8.1
Copyright (C) 2013 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
```

So far so good. (I hope this worked for you.) Next try this. Plug a USB flash drive into your computer, and let Windows assign it a drive letter. Now the command "g++ --version" might cause a message like this to pop up:



The message persists even after going through the usual process of ejecting the drive. It also results from attaching certain other USB devices (such as a USB GPS device that installs partly as a flash drive). In my experience, it doesn't apply to USB hard drives present when the computer is started.

The first time I encountered this message, I found myself frustrated by not knowing why it had occurred. I didn't recall that I had installed or ejected a USB device, and I certainly didn't realize that such an event might have triggered the problem. I was trying to build the large package [ADMB](#), and I thought this message might indicate a bug in that software.

**Proposed solution:** Reboot your computer and don't install/eject USB devices during a session in which you plan to use MinGW. Also, don't boot your computer with a flash drive in a USB connector.

If someone finds a solution that doesn't involve rebooting, I'd be delighted to hear about it.

**What the blazes is going on?:** Currently, I have installed MinGW with the file "[mingw-get-setup.exe](#)", seemingly an "official" source. The description above reflects my own experience, based on many attempts with flash drives, a GPS unit, and computer reboots. I'm not certain that my proposed 'solution' will always work. For example, the real problem may relate to drive letters assigned to USB drives on my computer.

Other people have encountered this problem; for example <http://mingw.5.n7.nabble.com/mingw32-gcc-exe-No-Disk-td32484.html>. The difficulty might come from an inadvertent fixed reference to a drive letter on some developer's computer. One of the proposed solutions involves editing the executable files to make this reference relative, rather than absolute. The writer suggests: "*Use a strings editor to alter any broken paths in the final compiler executables, removing the drive spec so they begin with merely a forward slash.*" Personally, I'm not very happy with this development model. Why not fix the source code and rebuild the library, rather than attempt to alter the executable image? I suspect that this happens because the people who assemble the packages have no direct connection with the developers of component parts.

**Alternative software:** As an alternative, we could use [MinGW-w64](#). According to this website, the [R Project](#) uses this software, so presumably it can handle serious code. As I understand it, MinGW-w64 represents a complete break from the earlier [MinGW](#). According to the [Wikipedia article](#):

"Since the original MinGW project was not prompt on updating its code base, including the inclusion of several key new APIs and the much needed 64-bit support, [MinGW-w64](#) was created through a clean-room reimplementation originally authored by OneVision. OneVision then donated the code to the Public Domain in 2008. It was first submitted to the original MinGW project, but refused under suspicion of using non-public or proprietary information. For many serious reasons, the lead developer of the MinGW-w64 code decided to not to attempt further cooperation with MinGW."

I'm not alone in thinking that it's time to drop the original version of MinGW in favour of MinGW-w64. [Rioki's Corner](#) reaches the same conclusion and provides a helpful introduction: "MinGW-w64 for the Uninitiated".

I have written a report recommending support software for ADMB, based entirely on MinGW-w64. I'll send that separately.

Jon Schnute