

---

## [Why you should use the App Paths registry key \(http://www.sandon.it/node/84\)](http://www.sandon.it/node/84)

**Posted on:** Tue, 10/01/2013 - 15:05 **By:** Idsandon

Today François Piette has written about "[Path issue when having installed a lot of software \(http://francois-piette.blogspot.it/2013/10/path-issue-when-having-installed-lot-of.html\)](http://francois-piette.blogspot.it/2013/10/path-issue-when-having-installed-lot-of.html)": *Windows is having some trouble with very long path environment variable, even in Windows 7.* This is true. Although starting from Vista the *environment block size* is unlimited (environment variables are kept in a process environment, and each process gets a copy of the environment of the process launching it), the size of a *single environment variable* ([http://msdn.microsoft.com/en-us/library/windows/desktop/ms682653\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms682653(v=vs.85).aspx)) is limited to 32767 bytes.

That means the more installers adds their applications path to the system or user environment variable, the higher the chance it runs out of space. Is there a solution? Yes, there's an elegant one, although it looks many developers, even of well known software, doesn't know - or use - it.

Some years ago I wrote here about the [useful App Paths registry key \(https://www.sandon.it/?q=node/24\)](https://www.sandon.it/?q=node/24). It's useful because it allows for *per-application* paths. That *Path* value specified there is prepended to the application environment path. Thereby an application doesn't need to "pollute" the global path if it just needs its own private path settings. MSDN says: " We encourage application developers to use the **App Paths** subkey to provide an *application-specific path* instead of making additions to the global system path". Unluckily, it just encourages... it should enforce it a little more.

Thereby if your application uses DLLs, BPLs or other common resources stored in a common folder, or in a subfolder, it can easily add it in its own App Paths key instead of modifying the system or user path. It's far easier to clean up also when the application is uninstalled or upgraded.

Just a caveat: App Paths works only when a process is launched using ShellExecuteEx(). It means, for example, it doesn't work for services, as far as I know.

No need then, to setup old-fashioned batch files - just write your setups to [register applications with the system properly \(http://msdn.microsoft.com/en-us/library/windows/desktop/ee872121\(v=vs.85\).aspx\)](http://msdn.microsoft.com/en-us/library/windows/desktop/ee872121(v=vs.85).aspx). "Copy and execute" is good only for simple applications that do not require complex dependencies. More complex applications need a proper setup to make them well-behaved citizen of the host system. Windows has many features that help to write application that plays well with the system, the shell and other applications - take advantage of them (and look for them!)

If Embarcadero started to do it as well with Delphi would be great also - and remember Delphi itself can modify the environment of processes it executes.

BTW: eventually *UseUrl* was documented: it allows an application to receive an URL when invoked

instead of having Windows download the URL content locally and pass this copy to the application.

- [Log in \(http://www.sandon.it/user/login?destination=/comment/reply/node/84/comment\\_node\\_blog%23comment-form\)](http://www.sandon.it/user/login?destination=/comment/reply/node/84/comment_node_blog%23comment-form) to post comments

---

**Source URL:** *http://www.sandon.it/node/84*