

Re: HowTo: Unregister a DLL/Control when the File no longer Exists

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2006-06/msg00932.html>

- *From:* "Tony Proctor" <tony_proctor@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 14 Jun 2006 09:07:42 +0100
-

We also have another internal utility that builds a large suite of VB projects, in the correct order, and deals with this exact situation Karl (i.e. unregistering the previous binary before letting VB6.exe generate the new one. It also does lots of other stuff too (e.g. automatically building locale-dependent resources) but that's OT. However, the RegClean type utility I mentioned before was for a different case. It's oh so easy to go and build a temporary copy of something, or install an older product version to test something, and then get lazy and just delete the directory afterwards – I've even done it myself. I guess you could put it down to a general COM pitfall. The only solution I can think of would be if the Delete operation first checked whether the file had self-registration code (which is quite easy) and then confirm quickly whether it was still recorded in the registry.

Tony Proctor

"Karl E. Peterson" <karl@xxxxxxxx> wrote in message <news:uWV1stxjGHA.3588@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

I wonder if an add-in could be written that automatically unregistered a axComponent before compiling one destined to overwrite it? That's my most-common scenario, when I'm in the very early (ie, design-by-coding) phases of a project.

--

Working without a .NET?
<http://classicvb.org/>

Tony Proctor wrote:

We wrote an internal RegClean type utility to deal with this situation since almost anyone can be guilty of deleting registered files (e.g. QA/Installation teams as well as developers). We found it was happening to a significant extent, plus Microsoft's RegClean didn't always fix it, plus a third party tool had already caused "registry bloat" by not cleaning the registry properly over a period of time.

Re: HowTo: Unregister a DLL/Control when the File no longer Exists

In principle, it's a matter of associating the Typelib records (may be multiple versions) with their respective ClsID records, IID records, and AppID records....and then deleting them all together. However, matching them up without the original DLL (or EXE, or TLB) is not foolproof. See

<http://groups.google.ie/group/microsoft.public.vb.com/msg/6b5dd45f89c04a0b?hl=en>

for instance

Tony Proctor

"Charles Law" <blank@xxxxxxxxxxxx> wrote in message news:#PHoB26iGHA.3408@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

I have a project group, in a specific location on disk, that contains several projects. There is the main EXE and a few ActiveX DLLs.

The main project references one of the DLLs, which in turn references another DLL. So far so good.

Periodically, I copy the entire directory hierarchy to another, similar, location, and call it version 2; so the parent directory MyProject still exists, but now there is one alongside it called MyProject #2. I begin working on #2.

When I am happy with the changes in #2, I delete the original directory structure altogether.

When I look in the references, I can still see the original controls. How can I remove them without wading through the registry in the vain hope that I know what I am doing? I realise that I can never include one of these as a reference because I get an "Error loading DLL" message, but in time I get a proliferation of entries in the references section to files that do not exist, and at best this creates clutter but at worst it means that I have to work through them until I find the right one to include. [Of course, until I delete the earlier version, it is still possible to include

Re: HowTo: Unregister a DLL/Control when the File no longer Exists

the wrong file, which is more serious]

TIA

Charles

.