How to get rid of the BDE

> ***QUESTION***
>
> I have (mostly) converted an application from Paradox to DBISAM. How do I know if I have really got the BDE out of my program?
>
> I have 150 forms in the application and I have been through them all fairly methodically, but how can I be sure that I'm not compiling in bits of BDE every time?
>
> Is there a thing in the "uses" clause that is there for BDE and not for DBISAM? Is there a way of looking at the EXE to see if there is BDE stuff in it?

Here is a list of possible approaches:

- Build the application with a MAP file (under linker options) and examine the map file looking for `DBTABLES`, `BDE`, or `BDECONST`. If any of these are included you're using the BDE. Having `DBTABLES` in a uses clause anywhere in the program is the most likely reason for the BDE to be used.
- Another option would be to remove or rename `DBTABLES.DCU` before doing a build all. That way the compiler will show you exactly where you are inadvertently using the BDE. Just searching unit files for particular units may not be good enough as you may be using third party components without source code that may themselves use the BDE.
- Look out for calls in the uses clause: *dbtables*, *bde* and *bdi*.
- The final killer is if you can test the app in a machine without BDE. If you want to do that in your own machine you can simply rename the `HKEY_CURRENT_USER/Software/Borland` key to something else (I use something like `xxBorland`) and `HKEY_LOCAL_MACHINE/Software/Borland` to something else also (as before, I use `xxBorland`). After renaming these keys in the registry the BDE is out and you can test your app in your machine. If it works, it'll be OK. Don't forget to rename back those keys in the registry to be able to use your BDE again.
- Rename the BDE directory. A lot easier than messing with the registry.

Original resource:	The Delphi Pool
Author:	Unknown
Added:	2013-01-28
Last updated:	2013-01-28

Copyright © Peter Johnson (*DelphiDabbler*) 2002-2018