

## RE: Registering a custom DLL after deployment – advice?

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-07/msg01846.html>

---

- *From:* Sergey Poberezovskiy <[SergeyPoberezovskiy@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:SergeyPoberezovskiy@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Sun, 15 Jul 2007 17:32:03 -0700
- 

Rich,

In the setup project/file system editor – this is where you need to set the Register property to vsdraCOM – and leave the output as content

"Rich" wrote:

Thank you. I tried making a reference to the com dll, but that did not register the dll on the destination workstation. Then I did as you say – add to the output. I added the dll and the tlb and set the build action to content and copy to output directory "copy Always". But that did not register the dll. Next I will try setting the build action to compile. If that doesn't register the dll then I will try deploying the .bat file on publishing and invoke the .bat file.

Hopefully, I can register the dll by setting the build action to compile.

Thanks again for your replies,  
Rich

"Sergey Poberezovskiy" wrote:

You do not have to reference your COM dll from console project if you do not use it there – in the setup project you can just add any file to the output.

"Rich" wrote:

Thank you for replying to my post.

I was thinking about p.WaitforExit()

But I like the idea of letting .Net do the deploying for me.

RE: Registering a custom DLL after deployment – advice?

Question: The custom dll is for the Access ADP (for com use). I make a reference to this dll in the Access ADP. So, in order to deploy it – do I make a reference to the dll in the console application so that it is included in the Application files? I am thinking yes. Well, I will try that tomorrow.

"Sergey Poberezovskiy" wrote:

Rich,

You do not have to use bat files – you can ask .Net setup & deployment project to do this for you – just add the tlb to the output (which I believe you already do), and set its Register property to vsdraCOM.

Anyhow, if you want to run a batch file and delete it after completion, you will need to get the started process reference and wait for its completion, similar to the following:  
p.Start();  
p.WaitForExit();

"Rich" wrote:

I use "Click Once" Deployment from VS2005. Works like a charm. I have to deploy a console app and an MS Access ADP. The console app invokes the ADP. I copy the ADP to the application files to be deployed from the console app and then use the Process object to invoke the ADP on the end User's

## RE: Registering a custom DLL after deployment – advice?

workstation. Below in this post is the code I use in the console application.

Anyway, I have had to add some additional functionality to the ADP in the form of a dll I wrote in .Net which uses .Net objects (sqlDataAdapter for reading data from Sql Server, StreamWriter for writing data to a Text File). The DLL works fine.

My plan is this: On deployment, I will deploy a copy of the DLL (and tlb) file(s) and a .bat file for registering the dll on the end user's workstation on first usage of the deployment. But I don't want to keep re-registering the dll everytime the enduser invokes the ADP (from the console app which will be a shortcut on the endusers workstation). So I am thinking I will delete the .bat file after first usage. Then, on the next deployment, the .bat file will be available again for usage. Here is the code:

```
using System;
using
System.Collections.Generic;
using System.Diagnostics;
using System.Text;
using System.IO;
namespace
SubscriberClientRnD
{
static class Program
{
/// <summary>
```

RE: Registering a custom DLL after deployment – advice?

```
/// The main entry point for
the application.
/// </summary>
[STAThread]
static void Main()
{
    //my version – as of
    11/17/06
    Process p = new Process();
    if
    (File.Exists("./RegisterEntireListDLL.bat")
    )
    {
        p.StartInfo.FileName =
        "./RegisterEntireListDLL.bat";
        p.Start();
        File.Delete("./RegisterEntireListDLL.bat");
    };
    p.StartInfo.FileName =
    "./subscriberRnD.adp";
    p.Start();
}
}
}
```

One question I have is if this is a good way to invoke the .bat file – using Process. Then, I am guessing that by invoking .bat this way – it is asynchronous and maybe I should use threading to make the app sleep for 500ms before deleting the file?

Any suggestions appreciated.

Thanks,  
Rich