

DLL vs Dynamic Code

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.general/2005-08/msg01591.html>

- *From:* John F <jf@xxxxxx>
 - *Date:* Tue, 30 Aug 2005 10:56:05 -0700
-

Currently I'm working on a C# app for a large organization (700 users nationwide). This App is still under development. I have setup code in my App to pull C# source code from a database and dynamically compile it at runtime. All source code stored in the database will be of Type Form and all forms will load as children of an MDI. Once I compile the code I store it on the local machine as a DLL. I do this so that the next time a user on that local machine elects to use a form, I check to see if that form exists locally in a DLL first. If it does I load it from the local DLL and cut down on both load time and network traffic. I also check to see if there are newer versions of the code in the DB and will pull/recompile as needed.

My question is: Can anyone give me pros/cons to storing either the C# code in the database or the already compiled DLL's? I could circumvent the need to compile on the fly if I stored the DLL's direct, but I'm not sure I recognize all the pros/cons to doing it either way. I do know that the DLL's are larger than the raw source code so network traffic would be up slightly. I think this would be nominal at best though.

Any thoughts/suggestion on storing raw source or a DLL would be greatly appreciated.

Security?
Speed?
Preference?

Thanks,
John F

-
- Prev by Date: [*RaiseError and VB .Net*](#)
 - Next by Date: [*System hook when Disk is Inserted?*](#)
 - Previous by thread: [*RaiseError and VB .Net*](#)
 - Next by thread: [*System hook when Disk is Inserted?*](#)
 - Index(es):
 - ◆ [*Date*](#)

◆ ***Thread***