Re: DLL vs Dynamic Compile

Source:
http://www.tech−archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005−08/msg05677.html

- *From*: "David Browne" <davidbaxterbrowne no potted meat@xxxxxxxxxxx>
- *Date*: Tue, 30 Aug 2005 13:37:42 −0500

"John F" <jf@xxxxxx> wrote in message
news:919E90BE−5DF6−41A3−B967−266F36A52F30@xxxxxxxxxxxxxxxx
> 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?
>

You are going to want to store the compiled binaries for all these reasons.

Speed and security right off, there are reasons to prefer binaries:

−A compiled assembly has an AssemblyVersion burned into it, while source code does not.
−A compiled assembly can be compiled from any number of source files.
−A compiled assembly can be signed, and loading compiled, signed, safe assemblies can be done in a low−trust context.

David

.

- ***Follow−Ups***:
  - ♦ ***Re: DLL vs Dynamic Compile***
    - ◊ *From:* John F

- ***References***:
  - ♦ ***DLL vs Dynamic Compile***
    - ◊ *From:* John F

- Prev by Date: ***RE: DLL vs Dynamic Compile***
- Next by Date: ***Re: Selecting appropriate override***
- Previous by thread: ***RE: DLL vs Dynamic Compile***
- Next by thread: ***Re: DLL vs Dynamic Compile***
- Index(es):
  - ♦ ***Date***
  - ♦ ***Thread***