

Forum:[Hack Forums](#)

Thread: [Hi guys. I'm trying to code a system where you can import a C# class \(.cs\) into a C#](#)

grweh3wrh - October 20, 2019, 3:50 pm

Hi guys,

I'm trying to code a system where you can import a C# class (.cs) into an existing C# program and interact with it programmatically.

All the classes that will be imported have the same method names, variable names and every name except the actual content of those methods, variables etc. Does anyone have any pointers as to where I could get started to make this?

Thanks!

smutnagab - October 20, 2019, 4:03 pm

If you're referring to run-time JIT compilation for C# then you would need to use the .NET JIT compiler manually to produce byte-code for the C# classes you are trying to compile at run-time and then would need to figure out how to run that byte-code from memory.

0xDEAD10CC - October 20, 2019, 11:19 pm

CodeDOM would be the traditional approach if you're looking to use something provided within the .NET framework.

MavenGradle - October 21, 2019, 2:20 am

(Oct 20, 2019 11:19 PM)0xDEAD10CC Wrote: [\[CITE\]](#) CodeDOM would be the traditional approach if you're looking to use something provided within the .NET framework.

In addition to that, Reflection would then traditionally be used to load the assembly up into memory without ever writing to the disk (which in reality does not happen since files will still be written and read from disk during and after compilation using 'temporary files'), granted you utilize the correct parameters when using CodeDOM.

OP will have to unload said assemblies from the AppDomain after completion or risk running into several issues which is easier said than done, not to mention security issues. So ideally he would create a new AppDomain.

Noteworthy here is how as time goes on, new ways of doing this are introduced. Take Roslyn for example in combination with the SyntaxTree API. However, this does not solve the AppDomain issue and code integrity/security flaws that may be introduced when loading external modules.

I will not detail process injection because that is simply a malware tactic (or can be used as such) that can be avoided entirely.

0xDEAD10CC - October 21, 2019, 4:45 pm

Marshalling data between appdomains is relatively easy though. Most people who have never used appdomains or don't know what they are would simply just load the assembly with normal reflection into the current domain though.