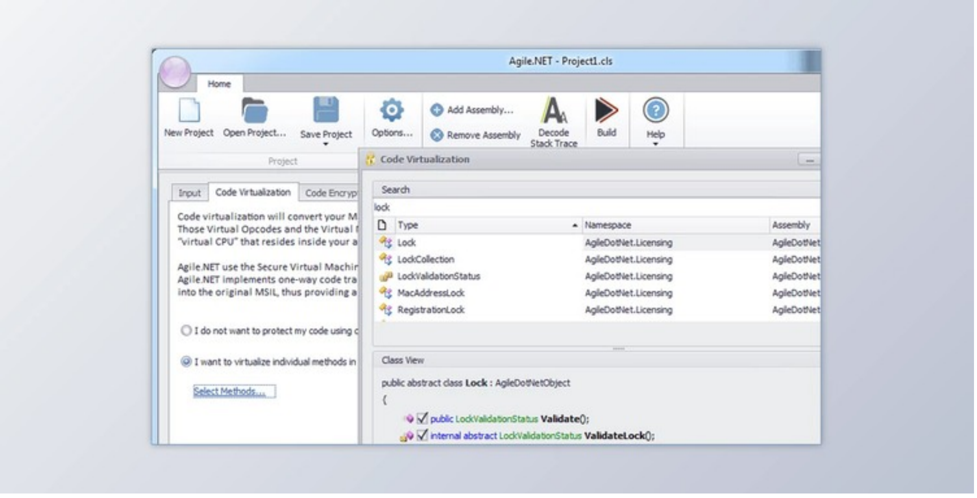
Agile.Net Obfuscator v6.6.0.12 + CRACK

2024-12-27 08:36:05 label 我要反馈 下载页面



Code virtualization will convert your M
Those Virtual Opcodes and the Virtual
"virtual CPU" that resides inside your a

Agile.NET use the Secure Virtual Machir
Agile.NET implements one-way code tra
into the original MSIL, thus providing a

○ I do not want to protect my code using c

● I want to virtualize individual methods in

lock

Type	Namespace	Assembly
Lock	AgileDotNet.Licensing	AgileDotNet
LockCollection	AgileDotNet.Licensing	AgileDotNet
LockValidationStatus	AgileDotNet.Licensing	AgileDotNet
MacAddressLock	AgileDotNet.Licensing	AgileDotNet
RegistrationLock	AgileDotNet.Licensing	AgileDotNet

```
public abstract class Lock : AgileDotNetObject
{
    ☑ public LockValidationStatus Validate();
    ☑ internal abstract LockValidationStatus ValidateLock();
```

Agile.NET Code Protection can help you obfuscate each component of your code, such as course and method names, managed tools, user strings, approach execution, system and library calls. It protects your program since it understands what's safe to change and what has to be left alone. Regardless, it gives you complete control of the obfuscation procedure.

Agile.NET obfuscator goes past conventional obfuscation procedures. Besides, renaming your metadata entities supports innovative obfuscation techniques that can harden your general protection scheme and transparency reverse technology entirely.

Agile.NET obfuscator renames all metadata constructs; this includes namespaces, class names, method signatures, and disciplines in addition to methods execution and chain values of your meeting. The renaming scheme comprises the 'unreadable chars' strategy; this technique will change courses, methods, and domain names to unprintable Unicode chars. When decompiled, the outcome is tough to comprehend source code. Because unprintable chars are used, it will not be possible to compile the resources generated after decompilation.

Agile.NET obfuscator offers control of stream obfuscation. Control flow obfuscation conceals the control flow data of this program by altering leaving code circulation patterns to semantically equivalent constructs, nevertheless different compared to the code initially composed. The control flow obfuscation algorithm transforms the initial implementation into spaghetti code, making it incredibly difficult to infer app logic. Agile.NET .NET obfuscator helps to ensure that program code circulation of the obfuscated assembly stays intact.

Cross Assembly Obfuscation enables the renaming of outside references, thus dramatically increasing the amount of obfuscated constructs. Provided a pair of assemblies that port every other, Agile.NET will rename classes, fields, and methods referenced from different assemblies uniformly. By way of instance, if class A announced in meeting A is referenced from meeting B and Agile.NET renames course A to A1, it's also going to rename B's external benchmark from A to A1.

Incremental obfuscation permits the programmer to make adjustments to the original resources after discharging an obfuscated meeting and then providing a patch into the consumer, reflecting the modifications to the initial program when maintaining the name-mapping employed from the first release. To achieve this, a map document has to be stored and later utilized to make sure that the renaming is maintained when creating changes and re-releasing that the obfuscated assembly.

An obfuscator must maintain the performance of the software completely undamaged whilst creating the initial source code unrecognizable when the obfuscated meeting is decompiled. Agile.NET obfuscator helps to ensure that the obfuscated meeting will operate in the same manner as the first assembly.

Obfuscation can pose problems when reflection API is employed at obfuscated assembly. Methods call You conducted via the use of expression API are very likely to fail when You obfuscated the program; this occurs because the procedure was renamed by the obfuscator, no matter how the telephone site still indicates the procedure its name. To mitigate these issues, Agile.NET obfuscator completely supports Microsoft's declarative obfuscation characteristics. All these features, announced right in the source code, enables the user to specify method and class names that shouldn't be uninstalled.

A frequent attacker will frequently search deployed assemblies for strings containing keywords like'GetLicense' or invalid License'. By discovering these strings, hackers try to bypass the license coverage embedded in the product which they're hacking. Agile.NET obfuscator provides the choice of series encryption.

Agile.Net Obfuscator Great Features:

- **Advanced obfuscation features for .NET platform -** Agile.NET obfuscator goes past conventional obfuscation procedures. Besides, renaming your metadata entities supports innovative obfuscation techniques that can harden your general protection scheme and transparency reverse technology entirely.

去下载

标签

.Net Components Other

- **Entity Renaming** - Agile.NET obfuscator renames all metadata constructs; this includes namespaces, class names, method signatures, and disciplines in addition to methods execution and series values of your meeting. The renaming scheme comprises the 'unreadable chars' strategy; this technique will change courses, methods, and domain names to unprintable Unicode chars. When decompiled, the outcome is tough to comprehend source code. Because unprintable chars are used, it will not be possible to compile the resources generated after decompilation.
- **Control Flow Obfuscation -** Agile.NET obfuscator offers control of stream obfuscation. Control flow obfuscation conceals the control flow data of this program by altering leaving code circulation patterns to semantically equivalent constructs, nevertheless different compared to code initially composed. The control flow obfuscation algorithm transforms the initial implementation into spaghetti code, making it incredibly difficult to infer app logic. Agile.NET .NET obfuscator helps to ensure that program code circulation of the obfuscated assembly stays intact.
- **Cross Meeting Obfuscation -** Cross Assembly Obfuscation enables renaming outside references, thus radically increasing the amount of obfuscated constructs. Provided a pair of assemblies that port every other, Agile.NET will rename classes, fields, and methods referenced from different assemblies uniformly. By way of instance, if class A announced in meeting A is referenced from meeting B and Agile.NET renames course A to A1, it's also going to rename B's external benchmark from A to A1.
- **Incremental obfuscation -** Incremental obfuscation permits the programmer to make adjustments to the original resources after discharging an obfuscated meeting and then providing a patch into the consumer, reflecting the modifications to the first application when maintaining the name-mapping employed from the first release. To achieve this, a map document has to be stored and later utilized to make sure that the renaming is maintained when creating changes and re-releasing that the obfuscated assembly.
- **Program Code Flow Remains Intact - an obfuscator must maintain** the performance of the software completely undamaged whilst creating the initial source code unrecognizable when the obfuscated meeting is decompiled. Agile.NET obfuscator helps to ensure that the obfuscated meeting will operate in the same manner as the first assembly.
- **Configuring your obfuscation procedure -** Obfuscation can pose problems when symptom API is employed at obfuscated assembly. Methods call You conducted via the use of expression API are very likely to fail when You obfuscated the program; this occurs because the procedure was renamed by the obfuscator, no matter how the telephone site still indicates the procedure its first title. To mitigate these issues, Agile.NET obfuscator completely supports Microsoft's declarative obfuscation characteristics. All these features, announced right in the source code, enables the user to specify method and class names that shouldn't be uninstalled.
- **String encryption -** A frequent attacker will frequently search deployed assemblies for sequences containing keywords like'GetLicense' or invalid License.' By finding such strings, hackers try to bypass the permit coverage embedded into the product which they're hacking. Agile.NET obfuscator provides the choice of series encryption.
- **X64 platform service -** Supports 32-bit and 64-bit software
- **Framework assistance -** Supports all variants of this. NET framework. You may also employ the obfuscation tool to safeguard applications installed beneath the. NET Compact Framework.
- **Mixed-mode Assemblies Service** - Agile.NET may obfuscate mixed-mode assemblies.
- **Debugging** - Among the side effects of obfuscation is that the problem of debugging obfuscated code. Exceptions developed and reported by a consumer will normally include obfuscated method and category names, making it nearly impossible to trace back the stack trace from the source code. Agile.NET obfuscator creates a clearly branded map document containing a comprehensive description of their obfuscated entities and their first titles; this advice is imperative to the consumer in translating debugger output by the obfuscated assembly.
- **MSBuild and NAnt build Integration -** Agile.NET incorporates with MSBuild and NAnt, thus boosting its use as part of a whole selection of complex build situations.

资源列表

download Agile.Net Obfuscator v6.6.0.12

inve 产品数量

已有 42647个

grou 付费会员

已有 1676位

anal 价值评估

商业价值约 ￥6635.87万元

dow 下载数量

已下载 222908次