

## Re: CodeDom V Emit with Dynamic Proxy

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.clr/2005-03/0161.html>

---

**From:** Sherif ElMetainy (*elmeteny.NOSPAM\_at\_wayout.net.NOSPAM*)

**Date:** 03/14/05

Date: Mon, 14 Mar 2005 23:46:13 +0200

Hello

Check out this tool

<http://www.thinktecture.com/Resources/Software/DynWsLib/default.html>

There is a problem though. Dynamically created proxies are assemblies that are loaded at runtime. .NET assemblies are never unloaded unless you unload the application domain. So if you have a client application that would be running for a long time (such as a web application), and the web service keeps changing, the client will end up having a lot of assemblies loaded that are not used anymore which affects performance.

I had a similar situation, and I created an application domain foreach web service client and marshaled the calls between my main app domain and the proxy's app domain. I unloaded the proxy's app domain when it is no longer needed (when my application got a notification that the configuration has changed).

Best regards,  
Sherif

"Tim Ellison" <btesubscriptions@direcway.com> wrote in message news:#0qNiy0JFHA.2428@TK2MSFTNGP10.phx.gbl...

Please point me in the correct direction for this type of discussion if I'm in the wrong newsgroup.

My problem domain is as follows:

A product we need to integrate with exposes a web service BUT its methods change as the product's repository changes. IOW, as new items are added to the repository, new methods are "added" to the web service, one for each item in the repository.

I have thought of two possible approaches to tackling this problem.

- 1) Learn how to write XML Soap messages really really fast (I'm just now confident with XPath).
- 2) Dynamically discover operations and types at run-time using WS Description and build an assembly using Emit or the CodeDom which will

contain the proxy.

I'm leaning towards (2) as the solution. Do you all know of any great books or URLs that concentrate solely on MSIL and/or CodeDom? Also, in your opinions, which direction is a better choice? Since this will be running under a web process, my security options are limited, but configuration will be a bear for the entire solution anyway, so what's one more permission setting (rhetorical). Thanks in advance.

--

TIM ELLISON