

## Dynamically Load WSDL Web Service

Written by Zack Mills  
Tuesday, 12 October 2010 08:07 -

---

Dynamically Load WSDL Web Service Code to try

<http://social.msdn.microsoft.com/Forums/en-US/csharpgeneral/thread/39138d08-aa08-4c0c-9a58-0eb81a672f54>

```
using System;

using System.Collections.Generic;

using System.Text;

using System.Reflection;

using System.CodeDom;

using System.CodeDom.Compiler;

using System.Security.Permissions;

using System.Web.Services.Description;

namespace ConnectionLib

{

    internal class WsProxy

    {

        [SecurityPermissionAttribute(SecurityAction.Demand, Unrestricted

            = true)]

        internal static object CallWebService(string webServiceAsmxUrl

            , string serviceName, string methodName, object[] args)

        {

            System.Net.WebClient client = new System.Net.WebClient();

            // Connect To the web service

            System.IO.Stream stream =

                client.OpenRead(webServiceAsmxUrl + "?wsdl");
```

## Dynamically Load WSDL Web Service

Written by Zack Mills

Tuesday, 12 October 2010 08:07 -

---

```
// Now read the WSDL file describing a service.

ServiceDescription description
= ServiceDescription.Read(stream);

///// LOAD THE DOM /////

// Initialize a service description importer.

ServiceDescriptionImporter importer
= new ServiceDescriptionImporter();

importer.ProtocolName = "Soap12"; // Use SOAP 1.2.

importer.AddServiceDescription(description, null, null);

// Generate a proxy client.

importer.Style = ServiceDescriptionImportStyle.Client;

// Generate properties to represent primitive values.

importer.CodeGenerationOptions =
System.Xml.Serialization.CodeGenerationOptions.GeneratePro
perties;

// Initialize a Code-DOM tree into which we will import
the service.

CodeNamespace namespace = new CodeNamespace();

CodeCompileUnit unit1 = new CodeCompileUnit();

unit1.Namespaces.Add(namespace);

// Import the service into the Code-DOM tree. This creates
proxy code that uses the service.

ServiceDescriptionImportWarnings warning =
importer.Import(namespace, unit1);

if (warning == 0) // If zero then we are good to go
{

// Generate the proxy code

CodeDomProvider provider1
```

## Dynamically Load WSDL Web Service

Written by Zack Mills

Tuesday, 12 October 2010 08:07 -

---

```
=CodeDomProvider.CreateProvider("CSharp");

// Compile the assembly proxy with the appropriate
references

string[] assemblyReferences = new string[5] ;

CompilerParameters parms
= new CompilerParameters(assemblyReferences);

CompilerResults results =
provider1.CompileAssemblyFromDom(parms, unit1);

// Check For Errors

if (results.Errors.Count > 0)

{

foreach (CompilerError oops in results.Errors)

{

System.Diagnostics.Debug.WriteLine("====Compiler
error====");

System.Diagnostics.Debug.WriteLine(oops.ErrorText)
;

}

throw new System.Exception("Compile Error Occured
calling webservice. Check Debug ouput window.");

}

// Finally, Invoke the web service method

object wsvcClass =
results.CompiledAssembly.CreateInstance(serviceName);

MethodInfo mi =
wsvcClass.GetType().GetMethod(methodName);
```

## Dynamically Load WSDL Web Service

Written by Zack Mills

Tuesday, 12 October 2010 08:07 -

---

```
        return mi.Invoke(wsvcClass, args);
    }
    else
    {
        return null;
    }
}
}
}
```

*More Code*

```
MethodInfo[] methods = service.GetMethods();
foreach (MethodInfo method in methods)
{
    if (isWebServiceMethod(method))
    {
        //Add in the list box or any con
```

## Dynamically Load WSDL Web Service

Written by Zack Mills

Tuesday, 12 October 2010 08:07 -

---

```
tainer
}
}
public static bool isWebServiceM
ethod(MemberInfo pMethodInfo) {
return (Attribute.GetCustomAttri
bute(pMethodInfo, typeof(System.
Web.Services.Protocols.SoapDocum
entMethodAttribute), false)!=nul
l ||
Attribute.GetCustomAttribute(pMe
thodInfo, typeof (System.Web.Ser
vices.Protocols.SoapRpcMethodAtt
ribute), false) != null);
}
```