

Recursive WebRequest.Create()

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2006-05/msg00135.html>

- *From:* "Sathyaish" <sathyaish@xxxxxxxx>
 - *Date:* 5 May 2006 03:04:28 -0700
-

The WebRequest class implements IWebRequestCreate and hence, a method Create.

This method has two other overloads, one of which is a private method. Here's how it looks:

```
public static WebRequest Create(string requestUriString)
public static WebRequest Create(Uri requestUri)
private static WebRequest Create(Uri requestUri, bool useUriBase)
```

Both the public overloads with a single argument internally call the private overload with the second parameter as false. The implementation of the private method is given below:

```
private static WebRequest Create(Uri requestUri, bool useUriBase)
{
    string text1;
    WebRequestPrefixElement element1 = null;
    bool flag1 = false;
    if (!useUriBase)
    {
        text1 = requestUri.AbsoluteUri;
    }
    else
    {
        text1 = requestUri.Scheme + ':';
    }
    int num1 = text1.Length;
    ArrayList list1 = WebRequest.PrefixList;
    for (int num2 = 0; num2 < list1.Count; num2++)
    {
        element1 = (WebRequestPrefixElement) list1[num2];
        if ((num1 >= element1.Prefix.Length) &&
            (string.Compare(element1.Prefix, 0, text1, 0, element1.Prefix.Length,
                true, CultureInfo.InvariantCulture) == 0))
        {
```

Recursive WebRequest.Create()

```
flag1 = true;
break;
}
}
if (!flag1)
{
throw new
NotSupportedException(SR.GetString("net_unknown_prefix"));
}
return element1.Creator.Create(requestUri);
}
```

I was surprised to see the last statement of this private overload calling one of the other public overloads. Isn't this recursive, or am I missing something here?