



Maverick

Articles

Contact

Sep  
20

## How To: Modify the web.config file in SharePoint using SPWebConfigModification 6

by Mark Wagner

### Summary

SharePoint 2007 (WSS and MOSS) allows you to easily and safely add and remove modifications to the web.config file. This post briefly covers what the SPWebConfigModification class can do for you, and how to best to use it.

I hope to improve this post with additions, corrections, and contributions from the public, so please provide any constructive comments and suggestions you have.

Technologies this applies to:

- Windows SharePoint Services 3.0 (WSS 3.0)
- Microsoft Office SharePoint Server 2007 (MOSS 2007)

### Acknowledgements

While I was scavenging for any information I could find regarding the SPWebConfigModification, there were a few posts I found useful. Information seemed to be scarce and I ended up learning most through trial and error; however, there a couple good nuggets and others I must acknowledge.

- **Shawn Feldman** is a coworker (very sharp developer) who was the first to make me aware of the SPWebConfigModification class. Additionally, he provided me with the original code samples, some of which I have since modified and included in my WebConfigurator

helper class (a separate post to come).

- **Vincent Rothwell** has a **post** entry that provides, in my opinion, a “Best Practice” on how to best remove your modifications using SPWebConfigModification. It’s simple and right on target.
- **Antonz** has a **post** entry that addresses the proper way to create a section that can be removed.
- **Daniel Larson** also has a few a **posts** that provide a basic overview and examples on how to use SPWebConfigModification.

## Overview of SPWebConfigModification

Through the use of the SPWebConfigModification class and xpath you can safely add and remove modifications to the web.config file. There are a number of benefits to using SPWebConfigModification to modify the web.config.

SharePoint provides access to a SharePoint web application via the object model using the SPWebApplication class. The SPWebApplication class has a SPWebConfigModifications collection property that contains all the modifications (SPWebConfigModification objects) made to this web application’s web.config file. This is also where we can add new modifications and have them applied to the web.config – or remove them. Additionally, you can iterate through this collection and inspect what changes have been applied. This contains all changes – including yours and those made by other assemblies. You can determine (or attempt to determine) where each change was made by inspecting the owner property.

**Every modification has an Owner.** Every change made the web.config can be assigned an owner. The owner is nothing more than a string value that uniquely identifies your changes from any other changes that have been made, or may be made in the future. I suggest using the name of the assembly, or the feature name or feature ID.

**Removing is not merely changing a value.** By assigning an owner to each new change you make, you can very easily remove your changes. In the case where your original modification was an addition,

where you added a new node or attribute, your changes are simply deleted and removed from the web.config file when they are removed. In the case where your modification changed the value of an existing node, upon removing your modification SPWebConfigModification automatically assigns the appropriate previous value. For example, let's say we wanted to change the Trust level in the web.config to "Full" from whatever it currently is. The Trust level is usually set to a value of "WSS\_Medium" by default, but it could easily be some other value before we make our changes. The great thing is that we don't need worry about what the current value is if ever we need to remove our modification. We do this by removing our change, not by trying to change the value back to a predetermined value such as "WSS\_Medium" – SPWebConfigModification manages this for us. This is a big advantage.

**Propagate your web.config changes across the farm.** SharePoint is designed to be very scalable and easily scalable. The changes you make to the web.config will need to be made to every front-end web server in the farm. SPWebConfigModification can manage this for you very easily.

Also, features are likely the method you will want to use to apply your changes. Using a feature (that can be either hidden or visible) will ensure your web.config changes are applied if a new web server is added to the farm, or rebuilt from scratch. Applying your web.config changes via a feature will ensure your changes are applied to new or newly built web servers – without having to refer to documentation or manually apply your changes to the web.config file on each and every web server.

## SPWebConfigModification properties

[SPWebConfigModification on MSDN](#)

### **Name**

Gets or sets the name of the attribute or section node to be modified or created. This is an XPath expression relative to the parent node (specified by the Path property). Sometimes this is nothing more than the name of the new or existing node you want to change. For example: "pages".

However, this often times needs to contain an XPath expression that is more specific in order to properly specify the correct node to modify (if it exists) or to create. For example:

```
"add[@assembly='myAssemblyName']"
```

### **Owner**

Gets or sets the owner of the web.config modification. This should to be a unique string value that will allow you to uniquely identify your modifications from any other modifications made to the web.config.

Suggestions are to use the assembly name, feature name, or feature ID. This will ensure you are able to easily and cleanly remove your modification when the need arises.

### **Path**

Gets or sets the XPath expression that is used to locate the node that is being modified or created. This value usually contains simple XPath expression to the parent node of the item you intend to change or create. For example: "system.web/httpHandlers" or "system.web/customErrors".

This can also be a more specific XPath expression to select (or create) a node with a specific attribute value. For example:

```
"configuration/configSections/sectionGroup[@name='mySection']"
```

### **Sequence**

Gets or sets the sequence number the modification. I have not performed any detailed testing with this parameter. I have always just set this value to zero (0) with no problems.

### **Type**

Gets or sets the type of modification for this object instance. The three values available are defined via the `SPWebConfigModification.SPWebConfigModificationType` and are **EnsureChildNode**, **EnsureAttribute**, and **EnsureSection**. Caution: Use the `EnsureSection` type with prudence as nodes created with `EnsureSection` cannot be removed. See Best Practice section below.

### **UpgradedPersistedProperties**

Gets the collection of field names and values for fields that were deleted or changed. I have not investigated this property and do not know of any need for using it. (inherited from `SPAutoSerializingObject`)

## Value

Gets or sets the value of the item to set. This is usually the complete xml element tag or attribute value for the node specified in the Name property (or as the name parameter in the constructor).

## How to use SPWebConfigModification

Using SPWebConfigModification is very easy, which makes it even better.

Below is a quick example of code that changes the *mode* attribute to a value of "Off". Here is a sample of what the web.config node might look like that this code would be modifying.

```
<configuration>
  <system.web>
    <customErrors mode="On">
  </system.web>
</configuration>
```

Here is the quick example that will change the *mode* attribute to "Off". Comments are inline describing each section of code.

```
private void SimpleSample()
{
    // Get an instance of my local web application
    SPWebApplication webApp = new SPSite("http://localhost");

    // Create my new modification to set the mode attribute to "Off"
    // Example: <customErrors mode="Off">
    SPWebConfigModification modification = new SPWebConfigModification
    modification.Owner = "SimpleSampleUniqueOwnerValue";
    modification.Sequence = 0;
    modification.Type = SPWebConfigModification.SPWebConfigModificationType.Modify;
    modification.Value = "Off";
    // Add my new web.config modification.
    webApp.WebConfigModifications.Add(modification);

    // Save web.config changes.
    webApp.Update();
    // Applies the list of web.config modifications to all Web app
    webApp.Farm.Services.GetValue<SPWebService>().ApplyWebConfigModifications(webApp.WebConfigModifications);
}
```

More examples can be found later in the article. Additionally, I will be posting my WebConfigurator helper class that contains a number of helper methods to centralize and simplify commonly modified elements in the web.config file.

## Best Practices

I found it difficult to fully understand exactly how to use SPWebConfigModification as well as how it should and should not be used, and how I should expect it for function. For this reason I have started a list of best practices when using SPWebConfigModification.

### **Saving your modifications**

While first using the SPWebConfigModification class, everything was going fairly smooth with testing and deployment – as it usually does on the developer machine. Of course my development machine is a single farm configuration, which is standard. Executing my web.config modifications on my single server worked just fine. However, when I deployed my code to two other farm configurations, the changes were not being applied to the web.config on any web server. Searching on the Internet surfaced a few other comments posted by others who were experiencing the same problem.

So, to correctly apply your changes on any farm configuration, be sure to perform the following two steps in your code:

1. Use the ApplyWebConfigModifications() method on the SPWebApplication object you are saving your SPWebConfigModification object to. I don't understand why this works over the other way, but it solved the problem for me. Example:

```
myWebApp.Farm.Services.GetValue<SPWebS
```

Note: This code did **not** work across the farm (I could not tell you why).

```
SPFarm.Local.Services.GetValue< SPWebS
```

2. Execute the Update() method on the SPWebApplication object. This will serialize the web application state and propagate changes across the farm – including your web.config changes.

```
myWebApp.Update();
```

So, whenever you want to save your modification to the web.config file, you should execute the following two statements.

```
// Save web.config changes.
myWebApp.Update();

// Applies the list of web.config modificat
myWebApp.Farm.Services.GetValue<SPWebService
```

### *Removing your modifications*

You should remove your modifications by checking the owner property on each SPWebConfigModification object in the SPWebConfigModifications collection. There are many examples that demonstrate removing modifications by creating a new SPWebConfigModification object and then removing that object. A must simpler, cleaner, and safer method is to remove your original SPWebConfigModification object by checking the Owner property on the SPWebConfigModification object. Credit goes to [Vincent Rothwell](#) for sharing this logic. Here is an example:

```
public void RemoveConfiguration(string owner)
{
    if (myWebApp != null)
    {
        Collection<SPWebConfigModification> collection = myWebApp.Farm.Services.GetValue<SPWebService>("SPWebConfigModifications");

        int iStartCount = collection.Count;

        // Remove any modifications that were
        for (int c = iStartCount - 1; c >= 0
        {
            SPWebConfigModification configMod
```

```

        if (configMod.Owner == owner)
            collection.Remove(configMod);
    }

    // Apply changes only if any items were
    if (iStartCount > collection.Count)
    {
        myWebApp.Update();
        myWebApp.Farm.Services.GetValue<S:
    }
}
}
}

```

### *Creating Sections*

#### *(EnsureSection vs. EnsureChildNode)*

When creating new section you should use the EnsureChildNode modification type value instead of the EnsureSection. This will allow you to remove your custom section when the need arises. [AntonZ](#) also discovered this as well and has an earlier post worth mentioning. Here is an example of the two options, both creating a section node like this: <mySection> </mySection>

This **cannot** be removed (using EnsureSection):

```

SPWebConfigModification mod = new SPWebConf
mod.Type = SPWebConfigModification.SPWebCon

```

This can be removed (using EnsureChildNode):

```

SPWebConfigModification mod = new SPWebConf
mod.Type = SPWebConfigModification.SPWebCo
mod.Value = "<mySection />";

```

### *Save Only Once per Event or Feature Activation*

It is important to remember that when developing and testing on a single server installation of SharePoint there are many activities that are able to complete instantly. However, when that same section of code executes on a SharePoint farm with multiple servers the time it takes to complete a given task is longer.

This is due to timer jobs and farm synchronization events that must occur.

For this reason, you should only execute the `ApplyWebConfigModifications()` method only once within a feature activation or event handler. If more than one call to the `ApplyWebConfigModifications()` method is made in close succession you will likely get the following error:

“ **A web configuration modification operation is already running.**

This is because request to save your changes from the first call to the `ApplyWebConfigModifications()` is not fully completed across the farm.

### *SPWebConfigModification Sample Feature*

I intend to post a blog entry containing code samples of my `WebConfigurator` class and a sample of how to implement `SPWebConfigModification` using a web application feature.

### Related Posts

`WebConfigurator` – a helper class to assist in using `SPWebConfiguration`.

### *References*

- Shawn Feldman is a coworker (and a very sharp developer) who was the first to make me aware of the `SPWebConfigModification` class. Additionally, he provided me with code samples, some of which I have since modified and included in my `WebConfigurator` helper class (a separate post to come).  
<http://feldrox.spaces.live.com/>
- Vincent Rothwell  
<http://blog.thekid.me.uk/archive/2007/03/20/removing-web-config-entries-from-sharepoint-using-spwebconfigmodification.aspx>
- Antonz  
<http://antonzhi.spaces.live.com/Blog/cns!76D1A388C70D6997!108.entry>
- Daniel Larson  
<http://daniellarson.spaces.live.com/blog/cns!D3543C5837291E93!958.entry>

Posted in *.NET, Development, SharePoint*

Tagged *Development, SharePoint*



*Mark Wagner*

— SEPTEMBER 27, 2010 AT 9:38 AM

0 0 Rate This

(Migrated Comments from my old blog)

Best Practices for making multiple changes

If I had multiple changes to my web.config what would you consider the best practice for structuring my feature to make all the changes at one time?

-John

at 10/9/2007 1:42 PM

Best Practices for making multiple changes

John, I'm not sure if I completely understand your question. But here would be my general approach.

Making many changes to the web.config in a single feature is very natural and acceptable to do. I don't see a need to have to structure your feature(s) to meet the needs of modifying your web.config. Design your features in a manner that is best for your feature(s). When making many changes, just be sure not to call the ApplyWebConfigModifications method more than once within a single feature activation. You can still add 2, 10, or even 30 changes to the web.config; just be sure to call the ApplyWebConfigModifications once at the end.

Mark at 10/13/2007 9:30 AM

Update

I was just looking for an example where someone was making more than one update. Understanding how to use ApplyWebConfigModifications to actually modify what I wanted was challenging enough :-)

Ted Pattison has a fantastic post today with a code example where he makes several modifications to the web.config. Great example to go along with your info!

<http://blog.tedpattison.net/Lists/Posts/Post.aspx?List=80e51818%2D7b98%2D41d8%2Db313%2D955977ecd68f&ID=4>

at 10/18/2007 8:20 AM

Making changes in a particular order.

I am trying to add the changes necessary for using AJAX Extensions to the web.config using your method. I run into a problem with the changes not being in the order that I add them in. I've tried adding them in the collection in the correct order, and also using an incrementing sequence number for the SPWebConfigModification object. But no matter how I do it, it ends up in the same order. For example, I get this:

Instead of this:

Even though I add them using the right order and sequence. Any suggestions?

Thanks!

at 11/6/2007 10:44 AM

Re: Making changes in a particular order.

Sorry about the delay. I just noticed this comment. I don't get notified when someone posts a comment and I need to set that up to prevent this type of delay.

As the for problem you are encountering, this is the first I have seen this – and it may be a known problem with SPWebConfigModification. I don't have an answer at this time, but I appreciate you posting this comment and letting my know. I will perform some testing myself to see if there is anything I come up with – and will let you know if I do find a resolution.

Mark at 11/10/2007 8:03 AM

Re: Making changes in a particular order.

Mark,

Thanks for responding to my question above about "Making changes in a particular order". I did some more testing on the issue of the modifications not being added in the correct order. The MSDN says "Lower sequence number modifications are applied first." (<http://msdn2.microsoft.com/en-us/library/microsoft.sharepoint.administration.spwebconfigmodification.sequence.aspx>) But adding different sequence numbers doesn't seem to work.

I found that the configuration changes are stored in the Objects table of the Config database. If you do:

```
SELECT * FROM Objects WHERE Name LIKE  
'WebConfigChanges%'
```

You'll easily find the changes. I noticed that they are stored in the database out of sequence. Instead it seems like they are stored in the database in descending order based on the SPWebConfigModification.Name property.

```
So for the the following  
SPWebConfigModification.Name and  
SPWebConfigModification.Value:  
Name = "remove[@sequence='" + sequence + "']"  
Value = ""
```

```
Name = "add[@sequence='" + sequence + "']"  
Value = ""
```

The "add[@path='\*.asmx']" will always come before the "remove[@path='\*.asmx']" even if I add the "remove[@path='\*.asmx']" to the collection first and give it a lower sequence number.

I also played around with adding an extra "sequence" attribute to each change. Since I incremented the sequence number for each change, I could use this as the key to finding the node. This helped order things in the correct sequence for items that all have the same tag, but still didn't fix the ordering for items with different tag names. Plus, adding this extra attribute isn't the cleanest thing.

For example, I did the following for the Name and Value properties of each modification (where sequence was a uint that I incremented for each modification that I was making):

```
Name = "remove[@sequence='" + sequence + "']"  
Value = ""
```

```
Name = "add[@sequence='" + sequence + "']"  
Value = ""
```

I hope this helps in your research... maybe you'll find something that I overlooked.

Thanks,  
-Joe Bailey  
at 11/12/2007 11:08 AM

Re: Making changes in a particular order.

Meant to write that the changes are stored in ASCENDING order in the database based on the Name property...

-Joe

at 11/12/2007 11:11 AM

Hard coded values

I have added the required Ajax extension in my web.config. But i don't want hard coded values in my code, I want to move all of this hard coded things into another file. Do you have any suggestions?

at 12/26/2007 4:27 AM

Changing values depending on environment

Maybe it's in here somewhere and I just overlooked it - if that's the case, please accept my apologies.

But what if you want to update the values in the web.config file depending on the environment (dev., test, production)? What is the best approach? Is it for example possible to maintain the values in a separate XML file and have the operations team update these values or what would the best practise be?

at 1/9/2008 1:31 AM

Great!

Excellent post Mark!! Please do create a SPWebConfigModification Sample Feature and post it, the SharePoint community needs it! :-)

If you are looking for a real life example; maybe you can build a feature to get the ASP.NET AJAX modifications in a SharePoint web.config ... :-)

Regards,

Jan

<http://weblogs.asp.net/jan>

at 3/11/2008 12:56 AM

RE: Changing values depending on environment

I have done this using Property elements in the feature.xml file and then iterated through the collection in the FeatureActivated code:

foreach (SPFeatureProperty prop in

properties.Definition.Properties)

{}

Cheers

Dave

at 4/14/2008 5:12 PM

Re: Making changes in a particular order

I tried to make the changes with the sequence element, but with no results. Does anyone has done it already, so that person can share to us how she/he resolved the problem?

Thanks,

Adrian

at 4/16/2008 5:22 PM

what about removing nodes that are in the web.config?

Hello, great post, one thing that I dont cant figure out is how to remove nodes that I havent put in the web.config in the first place.

Say you want to remove the entire node.

I can change the mode to the value "InProc" for sure... (which is the sessionState that I want to use instead), but when I do that I get a message stating there are illegal attributes to that type of sessionState... So thats not an option.

And the same goes with removing attributes ... Say if i manage to change the mode to InProc... how do I remove other attributes that arent allowed on that type of sessionState?

Any thoughts/ideas on this ?

Cheers, Andy

at 6/4/2008 2:38 PM

Typo

In your sample code, there is a typo on line ~9.

It should read:

```
modification.Type =  
SPWebConfigModification.SPWebConfigModificationType.EnsureAttribute;
```

It's missing the letter 'S' at the start.

at 6/30/2008 12:15 AM

Making changes in a particular order

I have a work around using xpath conditions in the name attribute of the WebConfigModification. It works a treat.

<http://stephenkaye.blogspot.com/2008/07/master-webconfigmodifications.html>

at 7/17/2008 9:35 AM

Updating web.config file in a farm env

Thanks for the post,

I tried the same cose from the blog to update the enteries in the web.config file. It works well in single server environment. In my test environment there are two web front end server. In this case the piece of code doesn't work.

Any inputs on the issue.

Thanks

Amit

at 8/11/2008 8:17 AM

Re: Updating web.config file in a farm env

Amit,

I am currently using this in several farms and it is working just fine. Be sure you are executing the following statements to apply your updates.

```
myWebApp.Farm.Services.GetValue().ApplyWebConfigModifications();  
myWebApp.Update();
```

Mark at 8/11/2008 3:58 PM

Updating web.config file in a farm env (Questions)

Mark or Amit,

I am having a similar issue where the line below does not appear to be

```
working:myWebApp.Farm.Services.GetValue().ApplyWebConfigModifications();
```

Do either of you think it could be load balancer related? We are using host headers with several sites on port 80. Our load balancer is pretty simple and in our current set up we have to have entries in the host file so that we can access the host header url on the

machine. Just wondering if any of this could be the culprit.

There are no errors in the log and I've written to the log after the update indicating the update occurred.  
at 10/24/2008 3:03 PM  
other modifications are also removed though I am using owner property

When I deactivate my feature, I remove only those with owner I had specified, still other connection strings are also removed.  
What can be the reason for this behaviour. Also, it does not seem to always remove them.  
Is there a way we can apply only specific modifications as only of particular owner.  
at 12/15/2008 4:13 AM  
Exception when calling GetValue

When I make the call to get the SPWebService, I get an XmlException with the message:  
"Name cannot begin with the '<' character, hexadecimal value 0x3C. Line 1, position 2."

I am seeing this same exception on two different MOSS installations.

Here's the code:

```
SPWebApplication curWeb =  
(SPWebApplication)properties.Feature.Parent;
```

```
if ( curWeb != null )  
{  
// ...add mod entries to  
curWeb.WeConfigModifications...  
}
```

```
curWeb.Farm.Services.GetValue().ApplyWebConfigModifications();  
curWeb.Update();
```

That second to last line is where the exception occurs in the GetValue call, prior to the call to ApplyWebConfigModifications.

I've also used the following line to get the SPWebApp, but the result is the same:  
curWeb = new  
SPSite("http://localhost").WebApplication;

Any suggestions on what the problem might be or how I can fix it?

at 12/18/2008 11:44 AM

Via Powershell

Using this approach from powershell:

<http://www.harleygreen.com/8/section.aspx/5>

at 2/7/2009 10:43 PM

The HttpHandler positioning problem

I got that one to work with:

```
Name: add[@path="*.asmx"]
```

```
Path: configuration/system.web/httpHandlers
```

```
Value: @"
```

```
"
```

Notice how I look for the ADD verb but insert the REMOVE first. It will clump up a few on the way but it will always work properly.

Regards,

Francisco Aquino

at 2/12/2009 3:58 PM

One extra note on httpHandlers

Testing a bit more I noticed that you can create a modification with an empty value for the , playing along with my previous suggestion you wont get them repeated and everything will be in the right order,

Regards,

Francisco Aquino

at 2/12/2009 4:04 PM

Update the 12/CONFIG/Web.Config

Do you know of a way to set the 12/CONFIG/Web.Config file?

at 4/28/2009 12:16 PM

Multiple webapplications config update on Multiple WFE servers

Hi,

Thanks for your post.

However this solution works only for single web application. In case if its required to update it on multiple web applications (say some key value pairs), we get the error " A web configuration modification

operation is already running.”.

As you already mentioned, it might be because of the delay in timer jobs and farm synchronization events.

In such a scenario( updating multiple config's from the same event), what will be the best approach to go with ?.

at 4/30/2009 3:34 AM

Access Denied!

Hi, I'm trying to run the code when you activate a feature, but when it calls the `myWebApp.Farm.Services.GetValue().ApplyWebConfigModifications();` the code errors with an access denied. This works fine on my Development stand alone, but when testing in a Farm environment it fails every time. Do you have any ideas why?

at 4/30/2009 7:59 AM

Re: Access Denied

I'm not certain what access is required to call that statement, but anytime you need to execute code with elevated permissions, you can do so by wrapping your statement with the SharePoint `RunWithElevatedPrivileges` statement. Here is the URL showing an example of how to do this.

<http://www.crs.com/mark/Lists/Posts/Post.aspx?ID=43>

Mark Wagner at 5/1/2009 10:00 AM

Re: Multiple webapplications config update on Multiple WFE servers

Your assumption is incorrect. In this article I discuss when and why you may get this error, and how to prevent it. Please refer back to the article.

Mark Wagner at 5/1/2009 10:01 AM

Updates are currently disallowed on GET requests

I am trying to follow your example to update the web.config, but am running into an error. I have an Infopath form content type on a list. In the code of the form I am trying to implement updating the web.config. My snippet is below along with the error, which occurs on the line

```
WebApp.Farm.Services.GetValue().ApplyWebConfigModifications();
```

Any ideas where I am going wrong? thanks

```

public void UpdateWebConfig(ModificationEntry
myModEntry)
{
string mySiteUrl =
SPContext.Current.Web.ParentWeb.Url;
using (SPSite mySite = new SPSite(mySiteUrl) )
{
mySite.AllowUnsafeUpdates = true;
using (SPWeb myWeb = mySite.OpenWeb())
{
myWeb.AllowUnsafeUpdates = true;
myWeb.Update();
SPWebApplication WebApp = mySite.WebApplication;
SPWebConfigModification mod =
CreateModification(myModEntry);
WebApp.WebConfigModifications.Add(mod);
WebApp.Farm.Services.GetValue().ApplyWebConfigModifications();
WebApp.Update();
myWeb.AllowUnsafeUpdates = false;
}
mySite.AllowUnsafeUpdates = false;
}
}
}

```

Microsoft.SharePoint.SPException was unhandled by user code

Message="Updates are currently disallowed on GET requests. To allow updates on a GET, set the 'AllowUnsafeUpdates' property on SPWeb."

Source="Microsoft.SharePoint"

ErrorCode=-2130243945

StackTrace:

at

Microsoft.SharePoint.Library.SPRequest.ValidateFormDigest(String bstrUrl, String bstrListName)

at Microsoft.SharePoint.SPWeb.ValidateFormDigest()

at

Microsoft.SharePoint.Utilities.SPUtility.ValidateFormDigest()

at

Microsoft.SharePoint.Administration.SPPersistedObject.Update()

at

Microsoft.SharePoint.Administration.SPWebApplication.ApplyWebConfigModifications()

at

Microsoft.SharePoint.Administration.SPWebService.ApplyWebConfigModifications()

at

CMNewSiteForm.FormCode.c\_\_DisplayClass9.b\_\_8()

at

Microsoft.SharePoint.SPSecurity.CodeToRunElevatedWrapper(Object state)

at

```
Microsoft.SharePoint.SPSecurity.c__DisplayClass4.b__2()
at
Microsoft.SharePoint.Utilities.SecurityContext.RunAsProcess(CodeToRunElevated
secureCode)
InnerException:
System.Runtime.InteropServices.COMException
Message="Updates are currently disallowed on GET
requests. To allow updates on a GET, set the
'AllowUnsafeUpdates' property on SPWeb."
Source=""
ErrorCode=-2130243945
StackTrace:
at
Microsoft.SharePoint.Library.SPRequestInternalClass.ValidateFormDigest(String
bstrUrl, String bstrListName)
at
Microsoft.SharePoint.Library.SPRequest.ValidateFormDigest(String
bstrUrl, String bstrListName)
InnerException:
```

at 5/7/2009 2:43 PM

Re: Updates are currently disallowed on GET requests

You should allow unsafe updates (updates made via  
GET):

```
myWeb.AllowUnsafeUpdates = true;
mySite.AllowUnsafeUpdates = true;
```

Vitaly Mogoreanu

at 6/10/2009 6:40 AM

Re: Access Denied

I get the same error and did some research.

What I discovered is that in order to be able to apply  
the webconfig modification the application pool  
account running the update must be farm admin.

Basically, if you follow microsoft`s best practices for  
installation and set a different application pool account  
for each application pool, you will get an error when  
activating a feature with webconfig modifications.

A quick work around for this is to temporarily change  
the application pool account of the webapp from which  
you will activate the feature, to the same account as  
the central admin webapp. This will make everything  
work.

This is because the central admin application pool account has specific permissions on stored procedures on the sharepoint\_config database, which are required to run the update.

If anyone finds a way to apply those modifications without having to go through the security risk of changing the application pool account, please let me know!

at 7/9/2009 6:49 AM

"system.web/customErrors" was not found in the web.config file.

I tried the example of modifying web.config file ,but I am getting error on line  
webApp.Farm.Services.GetValue().ApplyWebConfigModifications();

Following is the error :

Failed to apply a web.config modification to file  
'C:\Inetpub\wwwroot\wss\VirtualDirectories\80\web.config'.  
The specified node "system.web/customErrors" was not found in the web.config file.

Can any one tell me what can be the problem?

at 7/9/2009 7:48 AM

Updating web.config changes after activating feature ...

In our Sharepoint farm, we do regular releases. If we want to change web.config values in different releases, should we stick to one feature and keep activating and deactivating it with the latest web.config modifications ,or should we have separate features everytime. Activating and deactivating feature appears to be the best solution, but wanted to know if there was any drawbacks to this approach

at 7/28/2009 11:56 AM

Re: Updating web.config changes after activating feature ...

From my experience, I see it as best to create a single "configuration" feature within each solution (.wsp) and simply reactivate that feature when deploying the updated/modified solution.

However, there may be times when a configuration feature is not always needed; and in this case it is best to make it a separate feature that can be activated/deactivated as needed. For example, if you had a feature that enabled or disabled ASP.NET debugging. You may only want that enabled when; lets

say in production, when you want error message details displayed temporarily.

But in general, I am of the opinion that a single configuration feature is best (for a given solution) as long as the configuration always needs to be applied to the web.config.

Mark Wagner at 7/29/2009 11:22 AM

Question on Remove

How the Webconfigmodifications.Count will be persisted. Let say I reset the IIS process. Next time the count will be zero aint it?

I have experienced same thing, what I have done wrong. What is your approach?

thanks in advance.

at 8/4/2009 2:25 AM

Re: Question on Remove

I'm not familiar with exactly what the Count property provides. I'm not sure if it is the current count of the current instance of the object or if it contains the total count of the the web config modification changes. I don't suspect an IIS reset will have any impact on that value. Hopefully someone else can answer this.

Mark Wagner at 8/5/2009 9:07 AM

EnsureSection vs. EnsureChildNode

If you use this code, after removing changes you'll get full section removed (imaging configuration\system.web\pages\controls). So all manual changes to the section are lost...Any suggestions?

```
SPWebConfigModification mod = new
SPWebConfigModification("mySection",
"configuration");
mod.Type =
SPWebConfigModification.SPWebConfigModificationType.EnsureChildNode;
mod.Value = "";
```

at 8/7/2009 2:00 AM

Not Removing Web.Config entires for Extended  
SharePoint Web Applications

I have created an easy way to add and remove changes for my web.config for my SharePoint web applications. When I activate my feature to modify the web.config it works fine.

When I deactivate my feature to restore my web.config back to its previous state it works fine except the web.config entries for any extended SharePoint web applications are NOT getting removed.

I am using the following code to perform the actual update.

```
WebApp.Farm.Services.GetValue().ApplyWebConfigModifications();  
WebApp.Update();
```

Any idea why this might be happening?  
at 9/22/2009 11:41 AM

[Reply](#)

---

Pingback: [I'm mentioned \(twice\) in an MSDN Magazine article – SWEET! « Mark Wagner's Cogitation Blog](#)

---



**Sam**

— FEBRUARY 12, 2011 AT 1:44 PM

0 0 Rate This

Hi Mark,

Great post.

I am using SharePoint 2010 and using the SPWebConfigModification in my FeatureActivated and FeatureDeactivating event within one of my features. I can deploy my custom page to sharepoint through my development environment but I am having issues when deploying it to a load balanced server environment. It gets deployed to all servers apart from one.

I experience the same message you described in your post: "A web configuration modification operation is already running."

I have tried all your code snippets but I can't seem to get my customisation to deploy in the load balanced SharePoint intranet.

I would be grateful if you could help.

Sam

[Reply](#)

---



**Ronak**

— MAY 18, 2011 AT 12:28 PM

0 0 Rate This

Thanks Mark for sharing knowledge and it is indeed very helpful.

Now i am trying to implement is whenever user add,update and delete item i want to corresponding action in web.config file for that entry is it something achievable ?

Thanks  
Ronak

[Reply](#)

---

Pingback: [Jumping the Hurdles of using SPWebConfigModification to Update the Web.config](#) « SPMatt

---



**Paul Ewert**

— JUNE 27, 2013 AT 11:56 AM

0 0 Rate This

Love that you covered the differences between single server and multi-server farms!

[Reply](#)

*Leave a Reply*

Enter your comment here...

← *WSPBuilder:*  
*SharePoint developers*  
– *Go Get It!*

*Fix: This connection*  
*requires an active*  
*Internet connection* →

---

SEARCH MY BLOG:

[.NET](#) [.Text](#) [Administration](#) [Apple](#) [ASP.NET](#)

[Aviation](#) [Books](#) [C#](#) [Debugging](#) [Development](#)  
[Governance](#) [Humor](#) [Internet Explorer](#) [Interview](#)  
[Questions](#) [JavaScript](#) [LiveMeeting](#) [Mac](#) [OS X](#)  
[Performance](#) [Remote Desktop](#) [Security](#) [Server](#)

[SharePoint](#) [Team Foundation Server](#)

[\(TFS\)](#) [Tools/Utilities](#) [Uncategorized](#) [Visual Studio](#) [Web](#)  
[Parts](#) [Web Services](#) [Windows](#)

[Maverick Limited](#) | [Maverick Assurance](#) | [Contact](#) | [Sign In](#) |  
[RSS](#)

Copyright 2001-2017 Mark Wagner



*"Simplicity is the ultimate sophistication."*

- Leonardo da Vinci

---

Create a free website or blog at [WordPress.com](#).