## **xdt:Locators**

**1.Match**

**In the provided syntax sample below the Replace transform will occur only when the name Northwind matches in the list of connection strings in the source web.config.Do note that Match Locator can take multiple attributeNames as parameters e.g. Match(name, providerName) ]**

**<connectionStrings>   
<add name="Northwind" connectionString="connectionString goes here"   
  
providerName="System.Data.SqlClient"   
  
xdt:Transform="Replace" xdt:Locator="Match(name)" />   
</connectionStrings>**

**2.Condition**

**Based on given condition , below transformation occurs**

**Original:- <add name="LocalMySqlServer" connectionString="dev server" providerName="System.Data.SqlClient" />**

**<connectionStrings>**

**<add name="Hello"**

**connectionString="mineserver"**

**xdt:Transform="Replace" xdt:Locator="Condition(@name='LocalMySqlServer' or @providerName='System.Data.SqlClient')"/>**

**</connectionStrings>**

**Result:-<add name="Hello" connectionString="mineserver" />**

3.**Xpath**

**<appSettings xdt:Transform="RemoveAll" xdt:Locator="XPath(//appSettings)">**

**...**

**</appSettings>**

**This code will remove appSetting where ever it is present in the web.config**

## **xdt:Transform**

**1.SetAttributes**

Takes the value of the specified attributes from the web.staging.config and sets the attributes of the matching element in the destination web.config. This Transform takes a comma separated list of attributes which need to be set. If no attributes are given to SetAttributes transform then it assumes that you would like to Set all the attributes present on the corresponding node in web.staging.config

<?xml version="1.0" encoding="utf-8"?>

<!-- For more information on using web.config transformation visit https://go.microsoft.com/fwlink/?LinkId=125889 -->

<configuration xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">

<!--

In the example below, the "SetAttributes" transform will change the value of

"connectionString" to use "ReleaseSQLServer" only when the "Match" locator

finds an attribute "name" that has a value of "MyDB".

-->

<connectionStrings>

<add name="LocalMySqlServer"

connectionString="mineserver"

xdt:Transform="SetAttributes" xdt:Locator="Match(name)"/>

</connectionStrings>

<system.web>

<compilation xdt:Transform="RemoveAttributes(debug)" />

<!--

In the example below, the "Replace" transform will replace the entire

<customErrors> section of your web.config file.

Note that because there is only one customErrors section under the

<system.web> node, there is no need to use the "xdt:Locator" attribute.

<customErrors defaultRedirect="GenericError.htm"

mode="RemoteOnly" xdt:Transform="Replace">

<error statusCode="500" redirect="InternalError.htm"/>

</customErrors>

-->

</system.web>

</configuration>

Result:- <add name="LocalMySqlServer" connectionString="mineserver" providerName="MySql.Data.MySqlClient" />

**2.Replace**

Completely replaces the first matching element along with all of its children from the destination web.config (e.g. staging environment’s web.config file). Do note that transforms do not modify your source web.config file.

<connectionStrings>

<add name="LocalMySqlServer"

connectionString="mineserver"

xdt:Transform="Replace" xdt:Locator="Match(name)"/>

</connectionStrings>

Result:- <add name="LocalMySqlServer" connectionString="mineserver" />

3.**Remove**

**Removes the first matching element along with all of its children**

**<connectionStrings>**

**<add name="LocalMySqlServer"**

**xdt:Transform="Remove" xdt:Locator="Match(name)" />**

**</connectionStrings>**

**It will remove below code from original web.config**

**<add name="LocalMySqlServer" connectionString="dev server" providerName="System.Data.SqlClient" />**

**4.RemoveAll**

**Removes all the matching elements from the destination’s web.config (e.g. staging environment’s web.config file).**

**Just as above this will remove all the matching elements named as LocalMySqlServer**

**<connectionStrings>**

**<add name="LocalMySqlServer"**

**xdt:Transform="RemoveAll" xdt:Locator="Match(name)" />**

**</connectionStrings>**

**5.Insert**

**Inserts the element defined in web.staging.config at the bottom of the list of all the siblings in the destination web.config (e.g. staging environment’s web.config file).**

**Original code:**

**<connectionStrings>**

**<add name="LocalMySqlServer" connectionString="dev server" providerName="System.Data.SqlClient" />**

**<add name="MySqlMembershipConnection" connectionString="Persist Security Info=False;server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb"**

**providerName="MySql.Data.MySqlClient" />**

**<add name="ConnectionStringLogger" connectionString="server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb"**

**providerName="MySql.Data.MySqlClient" />**

**</connectionStrings>**

**Transform code:-**

**<connectionStrings>**

**<add name="Insert" connectionString="Insert server" providerName="System.Data.SqlClient" xdt:Transform="Insert"/>**

**</connectionStrings>**

**Result:-**

**<connectionStrings>**

**<add name="LocalMySqlServer" connectionString="dev server" providerName="System.Data.SqlClient" />**

**<add name="MySqlMembershipConnection" connectionString="Persist Security Info=False;server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb" providerName="MySql.Data.MySqlClient" />**

**<add name="ConnectionStringLogger" connectionString="server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb" providerName="MySql.Data.MySqlClient" />**

**<add name="Insert" connectionString="Insert server" providerName="System.Data.SqlClient" /></connectionStrings>**

**6.RemoveAttributes**

**Removes the specified attributes from the destination web.config (i.e. staging environment’s web.config file).**

**<connectionStrings>**

**<add name="LocalMySqlServer"**

**connectionString="mineserver"**

**xdt:Transform="RemoveAttributes" xdt:Locator="Match(name)"/>**

**</connectionStrings>**

**Result:-**

**<connectionStrings>**

**<add />**

**<add name="MySqlMembershipConnection" connectionString="Persist Security Info=False;server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb" providerName="MySql.Data.MySqlClient" />**

**<add name="ConnectionStringLogger" connectionString="server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb" providerName="MySql.Data.MySqlClient" />**

**</connectionStrings>**

**The syntax example shows how multiple attributes can be removed.**

**<compilation   
  
xdt:Transform="RemoveAttributes(debug,batch)">   
</compilation>**

**7.InsertAfter (XPath)**

**Inserts the element defined in the web.staging.config exactly after the element defined by the specified XPath passed to “InsertAfter()” transform. In the syntax example the element <deny users="Vishal" />will be exactly inserted after the element <allow roles="Admins" /> in the destinationXML.**

**<authorization>   
<deny users="Vishal" xdt:Transform="InsertAfter(/configuration/system.web/authorization/allow[@roles='Admins'])" />   
</authorization>**

**OR**

**Original:-**

**<connectionStrings>**

**<add name="LocalMySqlServer" connectionString="dev server" providerName="System.Data.SqlClient" />**

**<add name="MySqlMembershipConnection" connectionString="Persist Security Info=False;server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb"**

**providerName="MySql.Data.MySqlClient" />**

**<add name="ConnectionStringLogger" connectionString="server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb"**

**providerName="MySql.Data.MySqlClient" />**

**</connectionStrings>**

**Transformation code:-**

**<connectionStrings>**

**<add name="InsertAfter" connectionString="Insert server after" providerName="System.Data.SqlClient" xdt:Transform="InsertAfter(/configuration/connectionStrings/add [@name='LocalMySqlServer'])" />**

**</connectionStrings>**

**Result:-**

**<connectionStrings>**

**<add name="LocalMySqlServer" connectionString="dev server" providerName="System.Data.SqlClient" />**

**<add name="InsertAfter" connectionString="Insert server after" providerName="System.Data.SqlClient" />**

**<add name="MySqlMembershipConnection" connectionString="Persist Security Info=False;server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb" providerName="MySql.Data.MySqlClient" />**

**<add name="ConnectionStringLogger" connectionString="server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb" providerName="MySql.Data.MySqlClient" />**

**</connectionStrings>**

**8.InsertBefore (XPath)**

**<connectionStrings>**

**<add name="InsertBefore" connectionString="Insert server Before" providerName="System.Data.SqlClient" /><add name="LocalMySqlServer" connectionString="dev server" providerName="System.Data.SqlClient" />**

**<add name="MySqlMembershipConnection" connectionString="Persist Security Info=False;server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb" providerName="MySql.Data.MySqlClient" />**

**<add name="ConnectionStringLogger" connectionString="server=devdb.cl7byd0npkno.ap-south-1.rds.amazonaws.com;user id=admin;password=mindit123;database=ajaydb" providerName="MySql.Data.MySqlClient" />**

**</connectionStrings>**

**Transformation code:-**

**<connectionStrings>**

**<add name="InsertBefore" connectionString="Insert server Before" providerName="System.Data.SqlClient" xdt:Transform="InsertBefore(/configuration/connectionStrings/add [@name='LocalMySqlServer'])" />**

**</connectionStrings>**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Cases1**

### Case 1: Replacing all AppSettings

**This is the overkill case where you just want to replace an entire section of the web.config. In this case I will replace all AppSettings in the web.config will new settings in web.release.config. This is my baseline web.config appSettings:**

**<appSettings>**

**<add key="KeyA" value="ValA"/>**

**<add key="KeyB" value="ValB"/>**

**</appSettings>**

**Now in my web.release.config file, I am going to create a appSettings section except I will include the attribute xdt:Transform=”Replace” since I want to just replace the entire element. I did not have to use xdt:Locator because there is nothing to locate – I just want to wipe the slate clean and replace everything.**

**<appSettings xdt:Transform="Replace">**

**<add key="ProdKeyA" value="ProdValA"/>**

**<add key="ProdKeyB" value="ProdValB"/>**

**<add key="ProdKeyC" value="ProdValC"/>**

**</appSettings>**

**Note that in the web.release.config file my appSettings section has three keys instead of two, and the keys aren’t even the same. Now let’s look at the generated web.config file what happens when we publish:**

**<appSettings>**

**<add key="ProdKeyA" value="ProdValA"/>**

**<add key="ProdKeyB" value="ProdValB"/>**

**<add key="ProdKeyC" value="ProdValC"/>**

**</appSettings>**

**Just as we expected – the web.config appSettings were completely replaced by the values in web.release config. That was easy!**

### Case 2: Replacing a specific AppSetting’s value

**Ok, so case 1 was the nuclear option, what about doing something a little more practical? Let’s go back to our original AppSettings web.config example:**

**<appSettings>**

**<add key="KeyA" value="ValA"/>**

**<add key="KeyB" value="ValB"/>**

**</appSettings>**

**Let’s say that we just want to replace KeyB’s value to something more suited for a production environment. This time we get to use both xdt:Transform and xdt:Locator.**

**Our strategy will be to define (in web.release.config) an appSettings section that has just the replacements we want to make. It will initially look like this:**

**<appSettings>**

**<add key="KeyB" value="ProdValA" />**

**</appSettings>**

**Now we want to add in the transforms so that we replace any appSetting that matches this key (KeyB).**

**<appSettings>**

**<add key="KeyB" value="ProdValA" xdt:Transform="Replace" xdt:Locator="Match(key)" />**

**</appSettings>**

**Once we publish our resulting web.config file looks like this:**

**<appSettings>**

**<add key="KeyA" value="ValA"/>**

**<add key="KeyB" value="ProdValA"/>**

**</appSettings>**

**Perfect – We replaced KeyB but left KeyA (and any other keys if they were to exist) alone.**

### Case 3: Replace Compilation Debug=”true”

**This is a simple one because Microsoft gives it to us out of the box – but I’ll reproduce it here anyway because it illustrates a common scenario and shows that there are more transforms then just Replace:**

**<system.web>**

**<compilation xdt:Transform="RemoveAttributes(debug)" />**

**</system.web>**

**There are also ways to SetAttributes, Remove elements, Insert elements, and much more**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Eg:**

**<appSettings> <add key="developmentModeUserId" value="00297022" /> <add key="developmentMode" value="true" /> /\* other settings here that should stay \*/ </appSettings>**

**into something like this:**

**<appSettings> <add key="developmentMode" value="false" /> /\* other settings here that should stay \*/ </appSettings>**

**<appSettings> <add key="developmentModeUserId" xdt:Transform="Remove" xdt:Locator="Match(key)"/> <add key="developmentMode" value="false" xdt:Transform="SetAttributes" xdt:Locator="Match(key)"/> </appSettings>**