

ASP.NET - Configuration

The behavior of an ASP.NET application is affected by different settings in the configuration files:

- machine.config
- web.config

The machine.config file contains default and the machine-specific value for all supported settings. The machine settings are controlled by the system administrator and applications are generally not given access to this file.

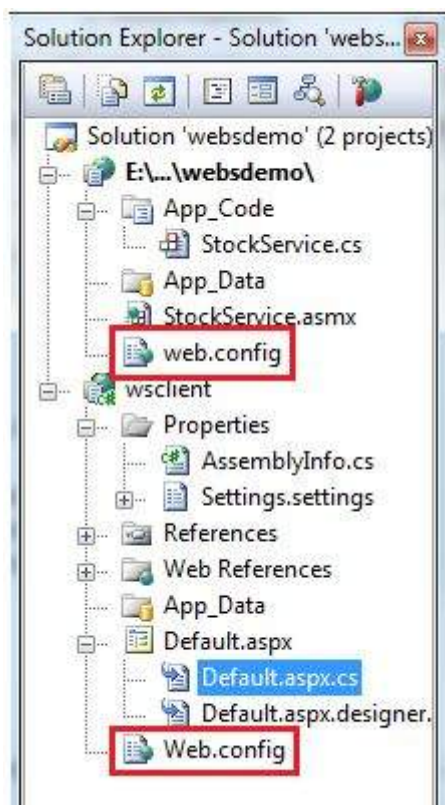
An application however, can override the default values by creating web.config files in its roots folder. The web.config file is a subset of the machine.config file.

If the application contains child directories, it can define a web.config file for each folder. Scope of each configuration file is determined in a hierarchical top-down manner.

Any web.config file can locally extend, restrict, or override any settings defined on the upper level.

Visual Studio generates a default web.config file for each project. An application can execute without a web.config file, however, you cannot debug an application without a web.config file.

The following figure shows the Solution Explorer for the sample example used in the web services tutorial:



In this application, there are two web.config files for two projects i.e., the web service and the web site calling the web service.

The web.config file has the configuration element as the root node. Information inside this element is grouped into two main areas: the configuration section-handler declaration area, and the configuration section settings area.

The following code snippet shows the basic syntax of a configuration file:

```
<configuration>

  <!-- Configuration section-handler declaration area. -->
  <configSections>
    <section name="section1" type="section1Handler" />
    <section name="section2" type="section2Handler" />
  </configSections>

  <!-- Configuration section settings area. -->

  <section1>
    <s1Setting1 attribute1="attr1" />
  </section1>

  <section2>
    <s2Setting1 attribute1="attr1" />
  </section2>

  <system.web>
    <authentication mode="Windows" />
  </system.web>

</configuration>
```

Configuration Section Handler declarations

The configuration section handlers are contained within the <configSections> tags. Each configuration handler specifies name of a configuration section, contained within the file, which provides some configuration data. It has the following basic syntax:

```
<configSections>
  <section />
  <sectionGroup />
  <remove />
  <clear/>
</configSections>
```

It has the following elements:

- **Clear** - It removes all references to inherited sections and section groups.
- **Remove** - It removes a reference to an inherited section and section group.
- **Section** - It defines an association between a configuration section handler and a configuration element.
- **Section group** - It defines an association between a configuration section handler and a configuration section.

Application Settings

The application settings allow storing application-wide name-value pairs for read-only access. For example, you can define a custom application setting as:

```
<configuration>
  <appSettings>
    <add key="Application Name" value="MyApplication" />
  </appSettings>
</configuration>
```

For example, you can also store the name of a book and its ISBN number:

```
<configuration>
  <appSettings>
    <add key="appISBN" value="0-273-68726-3" />
    <add key="appBook" value="Corporate Finance" />
  </appSettings>
</configuration>
```

Connection Strings

The connection strings show which database connection strings are available to the website. For example:

```
<connectionStrings>
  <add name="ASPDotNetStepByStepConnectionString"
    connectionString="Provider=Microsoft.Jet.OLEDB.4.0;
    Data Source=E:\projects\datacaching\ /
    datacaching\App_Data\ASPDotNetStepByStep.mdb"
    providerName="System.Data.OleDb" />

  <add name="booksConnectionString"
    connectionString="Provider=Microsoft.Jet.OLEDB.4.0;
```

```
Data Source=C:\ \databinding\App_Data\books.mdb"
providerName="System.Data.OleDb" />
</connectionStrings>
```

System.Web Element

The system.web element specifies the root element for the ASP.NET configuration section and contains configuration elements that configure ASP.NET Web applications and control how the applications behave.

It holds most of the configuration elements needed to be adjusted in common applications. The basic syntax for the element is as given:

```
<system.web>
  <anonymousIdentification>
  <authentication>
  <authorization>
  <browserCaps>
  <caching>
  <clientTarget>
  <compilation>
  <customErrors>
  <deployment>
  <deviceFilters>
  <globalization>
  <healthMonitoring>
  <hostingEnvironment>
  <httpCookies>
  <httpHandlers>
  <httpModules>
  <httpRuntime>
  <identity>
  <machineKey>
  <membership>
  <mobileControls>
  <pages>
  <processModel>
  <profile>
  <roleManager>
  <securityPolicy>
  <sessionPageState>
  <sessionState>
  <siteMap>
  <trace>
  <trust>
  <urlMappings>
```

```
<webControls>
<webParts>
<webServices>
<xhtmlConformance>
</system.web>
```

The following table provides brief description of some of common sub elements of the **system.web** element:

AnonymousIdentification

This is required to identify users who are not authenticated when authorization is required.

Authentication

It configures the authentication support. The basic syntax is as given:

```
<authentication mode="[Windows|Forms|Passport|None]">
  <forms>...</forms>
  <passport/>
</authentication>
```

Authorization

It configures the authorization support. The basic syntax is as given:

```
<authorization>
  <allow .../>
  <deny .../>
</authorization>
```

Caching

It Configures the cache settings. The basic syntax is as given:

```
<caching>
  <cache>...</cache>
  <outputCache>...</outputCache>
  <outputCacheSettings>...</outputCacheSettings>
  <sqlCacheDependency>...</sqlCacheDependency>
</caching>
```

CustomErrors

It defines custom error messages. The basic syntax is as given:

```
<customErrors defaultRedirect="url" mode="On|Off|RemoteOnly">
  <error. . ./>
</customErrors>
```

Deployment

It defines configuration settings used for deployment. The basic syntax is as follows:

```
<deployment retail="true|false" />
```

HostingEnvironment

It defines configuration settings for hosting environment. The basic syntax is as follows:

```
<hostingEnvironment idleTimeout="HH:MM:SS" shadowCopyBinAssemblies="true|false"
  shutdownTimeout="number" urlMetadataSlidingExpiration="HH:MM:SS" />
```

Identity

It configures the identity of the application. The basic syntax is as given:

```
<identity impersonate="true|false" userName="domain\username"
  password="<secure password>"/>
```

MachineKey

It configures keys to use for encryption and decryption of Forms authentication cookie data.

It also allows configuring a validation key that performs message authentication checks on view-state data and forms authentication tickets. The basic syntax is:

```
<machineKey validationKey="AutoGenerate,IsolateApps" [String]
  decryptionKey="AutoGenerate,IsolateApps" [String]
  validation="HMACSHA256" [SHA1 | MD5 | 3DES | AES | HMACSHA256 |
  HMACSHA384 | HMACSHA512 | alg:algorithm_name]
  decryption="Auto" [Auto | DES | 3DES | AES | alg:algorithm_name]
/>
```

Membership

This configures parameters of managing and authenticating user accounts. The basic syntax is:

```
<membership defaultProvider="provider name"
  userIsOnlineTimeWindow="number of minutes" hashAlgorithmType="SHA1">
  <providers>...</providers>
</membership>
```

Pages

It provides page-specific configurations. The basic syntax is:

```
<pages asyncTimeout="number" autoEventWireup="[True|False]"
  buffer="[True|False]" clientIDMode="[AutoID|Predictable|Static]"
  compilationMode="[Always|Auto|Never]"
  controlRenderingCompatibilityVersion="[3.5|4.0]"
  enableEventValidation="[True|False]"
  enableSessionState="[True|False|ReadOnly]"
  enableViewState="[True|False]"
  enableViewStateMac="[True|False]"
  maintainScrollPositionOnPostBack="[True|False]"
  masterPageFile="file path"
  maxPageStateFieldLength="number"
  pageBaseType="typename, assembly"
  pageParserFilterType="string"
  smartNavigation="[True|False]"
  styleSheetTheme="string"
  theme="string"
  userControlBaseType="typename"
  validateRequest="[True|False]"
  viewStateEncryptionMode="[Always|Auto|Never]" >

<controls>...</controls>
<namespaces>...</namespaces>
<tagMapping>...</tagMapping>
<ignoreDeviceFilters>...</ignoreDeviceFilters>
</pages>
```

Profile

It configures user profile parameters. The basic syntax is:

```
<profile enabled="true|false" inherits="fully qualified type reference"
  automaticSaveEnabled="true|false" defaultProvider="provider name">

<properties>...</properties>
<providers>...</providers>
```

```
</profile>
```

RoleManager

It configures settings for user roles. The basic syntax is:

```
<roleManager cacheRolesInCookie="true|false" cookieName="name"
  cookiePath="/" cookieProtection="All|Encryption|Validation|None"
  cookieRequireSSL="true|false" cookieSlidingExpiration="true|false"
  cookieTimeout="number of minutes" createPersistentCookie="true|false"
  defaultProvider="provider name" domain="cookie domain">
  enabled="true|false"
  maxCachedResults="maximum number of role names cached"

  <providers>...</providers>
</roleManager>
```

SecurityPolicy

It configures the security policy. The basic syntax is:

```
<securityPolicy>
  <trustLevel />
</securityPolicy>
```

UrlMappings

It defines mappings to hide the original URL and provide a more user friendly URL. The basic syntax is:

```
<urlMappings enabled="true|false">
  <add.../>
  <clear />
  <remove.../>
</urlMappings>
```

WebControls

It provides the name of shared location for client scripts. The basic syntax is:

```
<webControls clientScriptsLocation="String" />
```


WebServices

This configures the web services.
