

# Migration Guide

## Converting Applications from "Atlas" CTP to ASP.NET AJAX RTM

This document outlines the key steps that are required to migrate applications created with the ASP.NET "Atlas" CTP release to work with the Microsoft ASP.NET 2.0 AJAX Extensions 1.0 final (RTM) release. This document is intended to be a quick reference. For more details, please download the whitepaper [Changes between the ASP.NET AJAX \("Atlas"\) CTP and the RC and RTM Release](#) from the <http://ajax.asp.net> Web site.

Migrating an existing ASP.NET "Atlas" site to ASP.NET AJAX requires the following steps:

1. Installing the new release.
2. Modifying the application configuration (Web.config file).
3. Adding configuration settings for the AJAX Futures January CTP and ASP.NET AJAX Control Toolkit, if required.
4. Modifying existing ASP.NET Web pages.

Each of these steps is described below.

### 1. Install the new ASP.NET AJAX release

To install the new ASP.NET AJAX release, download it from the [ASP.NET AJAX Web site](#) and follow these steps.

1. Install the Microsoft ASP.NET 2.0 AJAX Extension 1.0 Final release.
2. Delete the Microsoft.Web.dll assembly from the Bin directory of your application. The core DLL is now installed in the .NET Framework global assembly cache (GAC).
3. Determine whether you need to install the ASP.NET AJAX Futures January CTP release. For a complete list of features that require the AJAX Futures January CTP, see the feature guide on the ASP.NET AJAX Web site.
4. If you are using controls from the "Atlas" Control Toolkit, install the new ASP.NET AJAX Control Toolkit. For information about changes, see the online documentation on the [Control Toolkit Web site](#).
5. If applicable, copy the new assemblies for the AJAX Futures January CTP and ASP.NET AJAX Control Toolkit into the Bin directory of your application. (These assemblies are not installed in the GAC.)

## 2. Modify the Web.config file

To use the new ASP.NET AJAX release with your existing application, you must update the Web.config file of the application. We strongly recommend that you do this by copying the sample ASP.NET AJAX Web.config file to your application and merging your application-specific settings into it. You can find the sample Web.config file at the following location:

```
%ProgramFiles%\Microsoft ASP.NET\ASP.NET 2.0 AJAX  
Extensions\v1.0.61025\Web.config
```

If you want to modify your Web.config file manually, use the steps detailed in the Appendix to this document.

## 3. Add configuration settings for the AJAX Futures January CTP and ASP.NET AJAX Control Toolkit

To add support for the AJAX Futures January CTP release to your application, add the following entries to the **<system.web><pages><controls>** section of your application's Web.config file:

```
<add namespace="Microsoft.Web.Preview.UI"  
  assembly="Microsoft.Web.Preview"  
  tagPrefix="asp"/>  
  
<add namespace="Microsoft.Web.Preview.UI.Controls"  
  assembly="Microsoft.Web.Preview"  
  tagPrefix="asp"/>
```

To add support for the ASP.NET AJAX Control Toolkit, see the instructions included with the Control Toolkit.

## 4. Modify existing ASP.NET Web pages

Migrating server components for use with ASP.NET AJAX is relatively straightforward, and mostly involves moving, renaming, and removing properties.

Server controls in ASP.NET AJAX now use **asp:** instead of **atlas:** as the namespace prefix. You can use the search and replace feature of your editor to rename server control tags to the **asp:** prefix.

The following is a list of changes you need to make to convert ASP.NET Web pages to use ASP.NET AJAX.

## ScriptManager and ScriptManagerProxy

- **EnablePartialRendering** is now set to **true** by default. If you relied on a different default setting, you must update your code.
- The **ErrorTemplate** element is no longer supported and must be removed. You can achieve the same functionality by setting the **AsyncPostBackErrorMessage** property to the default error message, or by handling the **AsyncPostBackError** method in JavaScript code. For more details, see the [whitepaper](#).
- Update Web service references to use the **<asp:>** tag prefix, as shown in the following example. In addition, change the **InlineProxy** property to **InlineScript**.

```
<Services>
  <asp:ServiceReference Path="~/webservicessample/helloworld.asmx" />
</Services>
```

- Update script references to use the **<asp:>** tag prefix:, as shown in the following example.

```
<Scripts>
  <asp:ScriptReference
    Name="Microsoft.Web.Resources.ScriptLibrary.PreviewScript.js"
    Assembly="Microsoft.Web.Preview" />
  <asp:ScriptReference
    Name="Microsoft.Web.Resources.ScriptLibrary.PreviewGlitZ.js"
    Assembly="Microsoft.Web.Preview" />
  <asp:ScriptReference Path="~/Scripts/Custom.js" />
</Scripts>
```

- **ScriptOnLoad** is no longer supported. Remove any references to this in your code.
- The **ScriptManager** and **ScriptManagerProxy** controls no longer have an **EnableScriptComponents** property. Remove references to this property in your code.
- Move settings that determine whether to render the debug or the retail version of JavaScript code out of the page and into the Web.config file, as shown here:

```
<configuration>
  <system.web>
    <compilation debug="false" />
  </system.web>
</configuration>
```

## UpdatePanel

- Change the **Mode** property to **UpdateMode**.
- Change the **IsUpdating** property to **IsInPartialRendering**.
- Change **ControlEventTrigger** to **AsyncPostBackTrigger**. Note that the **EventName** property on the trigger is now optional.
- Because the **ControlEventTrigger** feature of **UpdatePanel** was unreliable, it has been removed. To achieve a similar effect, use **AsyncPostBackTrigger** instead. The following examples illustrate this change.

### CTP

```
<asp:Button runat="server" id="Button1" Text="Go" OnClick="Button1_Click" />
<atlas:UpdatePanel ID="up1" mode="Conditional" runat="server">
  <ContentTemplate>
    <asp:calendar id="calendar1" runat="server" />
  </ContentTemplate>
  <Triggers>
    <atlas:ControlEventTrigger ControlID="Button1" EventName="Click" />
  </Triggers>
</atlas:UpdatePanel>
```

### ASP.NET AJAX version 1.0

```
<asp:Button runat="server" id="Button1" Text="Go" OnClick="Button1_Click" />
<asp:UpdatePanel ID="up1" UpdateMode="Conditional" runat="server">
  <ContentTemplate>
    <asp:calendar id="calendar1" runat="server" />
  </ContentTemplate>
  <Triggers>
    <asp:AsyncPostBackTrigger ControlID="Button1" EventName="Click" />
  </Triggers>
</asp:UpdatePanel>
```

## UpdateProgress

- The **UpdateProgress** control is now in the ASP.NET AJAX 1.0 core release.
- The **UpdateProgress** control no longer supports the functionality associated with a button whose ID is abortButton. To implement similar functionality, call the **PageRequestManager.abortPostBack** method in JavaScript. The following examples illustrate this change.

### CTP

```
<atlas:updateprogress runat="server" ID="Progress1" DisplayAfter="500">
  <ProgressTemplate>
    <b>working on request...</b>
    <input type="button" id="abortButton"
      value="Cancel" />
  </ProgressTemplate>
</atlas:updateprogress>
```

### ASP.NET AJAX version 1.0

```
<asp:updateprogress runat="server" ID="Progress1" DisplayAfter="500">
  <ProgressTemplate>
    <b>working on request...</b>
    <input type="button" id="abortButton" onclick="abortPB()"
      value="Cancel" />
  </ProgressTemplate>
</asp:updateprogress>

<script type="text/javascript" > function abortPB() {
  var obj = Sys.WebForms.PageRequestManager.getInstance();
  if (obj.get_isInAsyncPostBack()) {
    obj.abortPostBack();
  }
}
</script>
```

## TimerControl

- Change **TimerControl** to **Timer**. The following examples illustrate this change.

### CTP

```
<atlas:TimerControl runat="server"
  id="TimerControl1"
  Interval="60000"
  OnTick="TimerControl1_Tick"/>
```

### ASP.NET AJAX version 1.0

```
<asp:Timer runat="server"
  id="Timer1"
  Interval="60000"
  OnTick="Timer1_Tick"/>
```

## Extender Controls

- Extenders no longer have a properties section, and each extender control now maps to a single control. Update any extender markup accordingly. The following examples illustrate this change.
- The **AutoCompleteExtender** control is now part of the Control Toolkit instead of in the core release. The following examples illustrate this change.

### CTP

```
<asp:TextBox runat="server" ID="City" />
<atlas:AutoCompleteExtender ID="Ext1"
  runat="server">
  <atlas:AutoCompleteProperties
    TargetControlID="City"
    ServicePath="UIService.asmx" ServiceMethod="GetCities"
    MinimumPrefixLength="1" Enabled="true" />
</atlas:AutoCompleteExtender>
```

### ASP.NET AJAX version 1.0

```
<asp:TextBox runat="server" ID="City" />
<ajaxtoolkit:AutoCompleteExtender ID="autoComplete1"
  runat="server"
  TargetControlID="City"
  ServicePath="UIService.asmx" ServiceMethod="GetCities" />
```

## Appendix – Manual Configuration Changes

This section describes the changes you need to make to manually update your application configuration (Web.config) file to work with ASP.NET AJAX. The changes are described by section.

```
<configuration>
  <configSections>
```

- Replace the contents of this section with corresponding contents from the file located in the following folder:

```
%ProgramFiles%\Microsoft ASP.NET\ASP.NET 2.0 AJAX
Extensions\v1.0.61025\Web.config
```

```
<system.web>
  <compilation>
    <assemblies>
```

- Remove the reference to Microsoft.Web and add the following entry:

```
<add assembly="System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35"/>
```

```
<system.web>
  <pages>
    <controls>
```

- Remove the reference to the "atlas" tag prefix and add the following entries:

```
<add tagPrefix="asp" namespace="System.Web.UI"
assembly="System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
```

If you are using the ASP.NET Futures release, make sure that you keep the following two lines:

```
<add tagPrefix="asp" namespace="Microsoft.Web.Preview.UI"
assembly="Microsoft.Web.Preview"/>
<add tagPrefix="asp" namespace="Microsoft.Web.Preview.UI.Controls"
assembly="Microsoft.Web.Preview"/>
```

```
<configuration>
  <microsoft.web>
```

- Move the contents of this section to the following location:

```
<configuration>
  <System.Web.Extensions>
    <scripting>
      <webServices>
```

```
<configuration>
  <microsoft.web>
    <webServices>
      <profileService>
```

- Move the contents of this section to the following location:

```
<configuration>
  <System.Web.Extensions>
    <scripting>
      <webServices>
        <profileService>
```

- Set the correct value for the **enabled** property; by default it is not enabled.
- Change the **getProperties** attribute to **readAccessProperties**.
- Change the **setProperties** attribute to **writeAccessProperties**.

```
<configuration>
  <microsoft.web>
    <authenticationService>
```

- Move the contents of this section to the following location:

```
<configuration>
  <System.Web.Extensions>
    <scripting>
      <webServices>
        <authenticationService>
```

```
<configuration>
  <microsoft.web>
    <converters>
```

- Move the contents of this section to the following location:

```
<configuration>
  <System.Web.Extensions>
    <scripting>
      <webServices>
        <jsonSerialization>
          <converters>
```

You can also remove the existing entries.

```
<configuration>
  <System.Web>
    <httpHandlers>
```

- Update this section with the following changes to reflect the new type name, as shown in the following example:

```
<remove verb="*" path="*.asmx"/>

<add verb="*" path="*.asmx" validate="false"
  type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions,
  Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>

<add verb="*" path="*_AppService.axd" validate="false"
  type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions,
  Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>

<add verb="GET,HEAD" path="ScriptResource.axd"
  type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions,
  Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
  validate="false"/>
```

```
<configuration>
```

```
<System.Web>
```

```
<httpModules>
```

- Update this section with the following changes to reflect the new type name:

```
<add name="ScriptModule"  
type="System.Web.Handlers.ScriptModule, System.Web.Extensions,  
version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
```

```
<configuration>
```

```
<system.webServer>
```

- Add the following information:

```
<validation  
validateIntegratedModeConfiguration="false"/>
```

**Note** This section is used only with Microsoft Internet Information Server (IIS) 7.0.

```
<configuration>
```

```
<modules>
```

- Update this section with the following changes:

```
<add name="ScriptModule"  
preCondition="integratedMode"  
type="System.Web.Handlers.ScriptModule, System.Web.Extensions,  
version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
```

**Note** This section is used only with IIS 7.0.

```
<configuration>
```

```
<handlers>
```

- Update this section with the following changes:

```
<remove name="WebServiceHandlerFactory-ISAPI-2.0"/>
```

```
<add name="ScriptHandlerFactory" verb="*" path="*.asmx"
```

```
  preCondition="integratedMode"
```

```
  type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions,  
Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
```

```
<add name="ScriptHandlerFactoryAppServices" verb="*" path="*_AppService.axd"
```

```
  preCondition="integratedMode"
```

```
  type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions,  
Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
```

```
<add name="ScriptResource" preCondition="integratedMode"
```

```
  verb="GET,HEAD" path="ScriptResource.axd"
```

```
  type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions,  
Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
```

**Note** This section is used only with IIS 7.0.