**Using IIS Authentication with ASP.NET Impersonation**

**.NET Framework 4**

[Other Versions](javascript:;)

http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [Visual Studio 2008](http://msdn.microsoft.com/en-us/library/134ec8tc(d=printer,v=vs.90).aspx)
* [.NET Framework 3.0](http://msdn.microsoft.com/en-us/library/134ec8tc(d=printer,v=vs.85).aspx)
* [Visual Studio 2005](http://msdn.microsoft.com/en-us/library/134ec8tc(d=printer,v=vs.80).aspx)
* [.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/134ec8tc(d=printer,v=vs.71).aspx)

Internet Information Services (IIS) provides several authentication schemes that can be employed when securing a Web application. Common scenarios include using Integrated Windows authentication (NTLM) within a corporate intranet to determine application users' identity based on their Windows login, or specifying a single anonymous identity for a particular application. The Windows identity supplied by IIS can then be used to determine whether the Web application has access to a protected Windows resource, such as a file protected using an Access Control List (ACL), or a network resource such as a file or database server. You can configure ASP.NET to use the Windows identity supplied by IIS using impersonation.

By default, ASP.NET is configured to use Windows authentication mode, which applies the Windows identity supplied by IIS to the [User](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.user(v=vs.100).aspx) property of the current [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext(v=vs.100).aspx) object. This enables you to determine the identity supplied by IIS through the [User](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.user(v=vs.100).aspx) property (the user [Name](http://msdn.microsoft.com/en-us/library/system.security.principal.iidentity.name(v=vs.100).aspx) is blank when anonymous identification is used), but does not use the supplied identity as the [WindowsIdentity](http://msdn.microsoft.com/en-us/library/system.security.principal.windowsidentity(v=vs.100).aspx) for the current page. The [WindowsIdentity](http://msdn.microsoft.com/en-us/library/system.security.principal.windowsidentity(v=vs.100).aspx) for an application is used when determining if the application has access to a particular file or network resource.

To configure ASP.NET to impersonate the Windows identity supplied by IIS as the [WindowsIdentity](http://msdn.microsoft.com/en-us/library/system.security.principal.windowsidentity(v=vs.100).aspx) for the ASP.NET application, edit the Web.config file for the application and set the impersonate attribute of the identity configuration element to true, as shown in the following example.

<configuration>

<system.web>

<identity impersonate="true" />

</system.web>

</configuration>

Impersonation is independent of the authentication mode configured using the [authentication](http://msdn.microsoft.com/en-us/library/532aee0e(v=vs.100).aspx) configuration element. The authentication element is used to determine the [User](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.user(v=vs.100).aspx) property of the current [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext(v=vs.100).aspx). Impersonation is used to determine the [WindowsIdentity](http://msdn.microsoft.com/en-us/library/system.security.principal.windowsidentity(v=vs.100).aspx) of the ASP.NET application.

The following describes how you would enable impersonation using an intranet scenario as an example. In this scenario, you are setting up an internal corporate Web site for posting employee information. However, some of the information is for managers only. The manager information can be posted to a subdirectory of the employee information site, so that access to the information can be limited. IIS determines the user's identity using Windows Integrated (NTLM) security. The scenario assumes that:

* The Web server has the Microsoft Windows NT Server, Windows 2000 Server, or Windows Server 2003 operating system installed.
* IIS 6.0 is installed the Web server.
* The Web server hard disk is formatted using NTFS.
* All employees that need access to restricted resources are using Windows.

As the administrator of the application in the scenario, you would need to do the following:

1. Create the files and directories shown in the following illustration:

134ec8tc.directories(en-us,VS.100).gif

1. Create a Windows group named Managers that contains all users who should have access to the ManagerInfo.aspx file.
2. Use Internet Information Services (IIS) Manager to disable anonymous authentication for the application and enable integrated windows authentication.
3. In the application's Web.config file, set the impersonate attribute in the [identity](http://msdn.microsoft.com/en-us/library/72wdk8cc(v=vs.100).aspx) element to true.
4. Set the NTFS access control list (ACL) for the ManagerInformation directory to allow access to only those identities that are in the Windows Manager group and any required system accounts. You would need to be sure to include the identity of the ASP.NET process. The identity of the ASP.NET process for Windows 2000 Server or Windows NT is the local ASPNET account. The identity of the ASP.NET process for Windows Server 2003 and later is the identity of the IIS application pool, which by default is the NETWORK SERVICE account.

|  |
| --- |
| **NoteNote** |
| The ASP.NET role-management feature provides an alternative method of restricting access to areas of your Web application. For more information, see [Managing Authorization Using Roles](http://msdn.microsoft.com/en-us/library/9ab2fxh0(v=vs.100).aspx). |