**Windows Communication Foundation Authentication Service Overview**

The Windows Communication Foundation (WCF) authentication service enables you to use ASP.NET membership to authenticate users from any application that can send and consume a SOAP message. This can include applications that do not use the .NET Framework. Users of these different applications therefore do not need separate credentials for each application. Users can provide the same credentials when they use any one of the client applications, and be logged in to the application from all of them.

This topic contains the following sections:

* [Scenarios](http://msdn.microsoft.com/en-us/library/bb386582.aspx#Scenarios)
* [Background](http://msdn.microsoft.com/en-us/library/bb386582.aspx#Background)
* [Code Examples](http://msdn.microsoft.com/en-us/library/bb386582.aspx#CodeExamples)
* [Class Reference](http://msdn.microsoft.com/en-us/library/bb386582.aspx#ClassReference)

Scenarios

You access the authentication service as a WCF service when you have to authenticate users by using ASP.NET membership from an application that is not an ASP.NET Web application. This can include a console application, a Windows Forms application, or an application that is not developed with the .NET Framework. The application must be able to send and consume a SOAP message.

Background

To use the service, you pass the user's credentials to the authentication service, which validates the credentials by using ASP.NET membership. By default, the authentication service validates the user name and password by passing them to the default membership provider.

When the user has been authenticated, the ASP.NET authentication service issues an authentication ticket as an HTTP cookie that is compatible with ASP.NET forms authentication. In subsequent requests, the ticket is passed to the Web application so that the user does not have to provide credentials every time.

The authentication service does not support embedding the authentication ticket in the URL. Therefore, cookies must be enabled in the client to retain the authentication ticket.

For security reasons, the client application should always access the authentication service over the secure sockets layer (SSL, by using HTTPS protocol). For more information about how to set up SSL, see [Configuring Secure Sockets Layer (IIS 6.0 Operations Guide)](http://go.microsoft.com/fwlink/?LinkId=91844) and [Configuring Secure Sockets Layer in IIS 7.0](http://go.microsoft.com/fwlink/?LinkId=93377) on the Microsoft Web site.

**Authenticating with Custom Credentials**

You can add custom credentials in an authentication request when you have to validate the user by using information in addition to a name and password, such as an identification number. To include additional information for authentication, you pass the customized credentials in the *CustomCredential* parameter when you call the login method of the authentication service. You then create an event handler for the [Authenticating](http://msdn.microsoft.com/en-us/library/system.web.applicationservices.authenticationservice.authenticating.aspx) event. In the handler, you can read the credentials and validate them yourself. For more information, see [How to: Customize User Login When Using the WCF Authentication Service](http://msdn.microsoft.com/en-us/library/bb398862.aspx).

**Authenticating with a Custom Membership Provider**

You can authenticate the user name and password through a non-default membership provider by creating an event handler for the [Authenticating](http://msdn.microsoft.com/en-us/library/system.web.applicationservices.authenticationservice.authenticating.aspx) event. You then pass the user name and password to the [ValidateUser](http://msdn.microsoft.com/en-us/library/system.web.security.membershipprovider.validateuser.aspx) method of the custom membership provider. For more information, see [How to: Use Non-default Membership Provider for WCF Authentication Service](http://msdn.microsoft.com/en-us/library/bb386455.aspx).

You can retain non-sensitive user information as part of the authentication ticket. For example, you can store the user's favorite color in the ticket if you have to retrieve it later. To customize the authentication cookie, create an event handler for the authentication service's [CreatingCookie](http://msdn.microsoft.com/en-us/library/system.web.applicationservices.authenticationservice.creatingcookie.aspx) event and then store the user data in the [UserData](http://msdn.microsoft.com/en-us/library/system.web.security.formsauthenticationticket.userdata.aspx) property of the [FormsAuthenticationTicket](http://msdn.microsoft.com/en-us/library/system.web.security.formsauthenticationticket.aspx) object. For more information, see [How to: Customize the Authentication Cookie from the WCF Authentication Service](http://msdn.microsoft.com/en-us/library/bb398778.aspx).

**NoteNote:** Never store the user's password or other sensitive data in the ticket.