2011

COOKBOOK

Mixed Authentication

Code On Time web application generator supports ASP.NET Membership and several other authentication mechanisms. ASP.NET Membership is an attractive option for Internet applications and can be also successfully used in *intranet* applications deployed within network boundaries of an organization for use by a specific group of business users.

Web application administrator can use the advanced user manager provided with each generated application to create user accounts and manage roles.

Large organizations frequently mandate the need for a single sign-on mechanism to eliminate the need to manage multiple passwords and users accounts.

1. Typically a user name token is created and validated by the authentication software deployed to the local network. The user name token is embedded into each web request coming to a server. The authenticated user name can be found in a page request header variable.

2. Another option is to use the active directory identity name that can be available if Windows Authentication is enabled in your web application.

You can take advantage of either option to implement a mixed authentication based on the ASP.NET Membership option available in Code On Time database web application. Only the users registered in the ASP.NET Membership database of your application can access the application. User roles will also be derived from the membership database.

Users can self-register to use the application and will be able to access the application page when the user account is approved by administrator. Administrator can also create all authorized user accounts and assign the same “secret” password to all users.

Single sign-on is enabled through changes to the login user control. Open file *~/App_Code/Controls/Login.ascx* and modify the code-behind file as shown on the next page.

C#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Security;
using System.Security.Principal;

public partial class Controls_Login : System.Web.UI.UserControl
{

    protected void Page_Load(object sender, EventArgs e)
    {
        // Mixed authentication sample
        if (!Page.User.Identity.IsAuthenticated)
        {
            string userName = null;
            // 1. read the identity from the page request header variable
            userName = Request.Headers["UserName"];
            // 2. read the identity from the identity of the current Windows user
            userName = WindowsIdentity.GetCurrent().Name;
            // simulate the user name and ignore methods (1) and (2)
            userName = "admin";
            if (!String.IsNullOrEmpty(userName))
            {
                MembershipUser user = Membership.GetUser(userName);
                if (user != null)
                    FormsAuthentication.RedirectFromLoginPage(user.UserName, false);
            }
        }
    }
}
```

Methods of “silent” authentication are marked as (1) and (2). This particular example ignores the obtained information and simply assigns explicitly the user name “admin” to variable “userName”. Application makes a lookup request to identify the user as a valid ASP.NET Membership user. If that is the case then the user is automatically signed into the web application.

The login control is generated the first time only. Your changes to the code-behind file will stay intact with subsequent code generations.

You can adjust the sample to reflect your actual single sign-on method