



User privilege levels

```
using CMS.Membership;
using CMS.SiteProvider;

private bool CheckPrivilegeLevel()
{
    // Gets the currently authenticated user
    UserInfo user = MembershipContext.AuthenticatedUser;

    if (user != null)
    {
        // Checks whether the user has the Editor privilege level or higher
        return user.CheckPrivilegeLevel(UserPrivilegeLevelEnum.Editor, SiteContext.
CurrentSiteName));
    }

    return false;
}
```

Module permissions

```
using CMS.Membership;
using CMS.SiteProvider;

private bool CheckModulePermissions()
{
    // Gets the currently authenticated user
    UserInfo user = MembershipContext.AuthenticatedUser;

    if (user != null)
    {
        // Checks whether the user has the Read permission for the Content module
        return user.IsAuthorizedPerResource("CMS.Content", "Read", SiteContext.
CurrentSiteName);
    }

    return false;
}
```



Page type or custom table permissions

```
using CMS.Membership;
using CMS.SiteProvider;

private bool CheckPageTypePermissions()
{
    // Gets the currently authenticated user
    UserInfo user = MembershipContext.AuthenticatedUser;

    if (user != null)
    {
        // Checks whether the user has the Read permission for the CMS.MenuItem page
type
        return UserInfoProvider.IsAuthorizedPerClass("CMS.MenuItem", "Read",
SiteContext.CurrentSiteName, user);
    }

    return false;
}
```

Permissions for specific pages (ACLs)

```
using CMS.DocumentEngine;
using CMS.Membership;
using CMS.SiteProvider;
using CMS.DataEngine;

private bool CheckPagePermissions()
{
    // Creates a TreeProvider instance
    TreeProvider tree = new TreeProvider(MembershipContext.AuthenticatedUser);

    // Gets the Example page
    TreeNode page = tree.SelectSingleNode(SiteContext.CurrentSiteName, "/Example",
"en-US");

    if (page != null)
    {
        // Gets the user object
        UserInfo user = UserInfoProvider.GetUserInfo("CMSEditor");

        if (user != null)
        {
            // Checks whether the user has the Modify permission for the
Example page
            if (page.CheckPermissions(PermissionsEnum.Modify, SiteContext.
CurrentSiteName, user))
            {
                // Perform an action according to the result
                return true;
            }
        }
    }

    return false;
}
```

Filtering page data according to permissions

```
using CMS.Membership;
using CMS.Base;
using CMS.DocumentEngine;

...

// Gets the user object
UserInfo user = UserInfoProvider.GetUserInfo("CMSEditor");

// Gets all news items for which the user has Read permissions
using (new CMSActionContext(user))
{
    var news = DocumentHelper.GetDocuments("CMS.News")
        .OnSite("CorporateSite")
        .Path("/News", PathTypeEnum.Children)
        .CheckPermissions();
}
```