**To enable membership**

1. In Visual Web Developer, on the Web site menu, click ASP.NET Configuration.

The Web Site Administration Tool appears.

1. Click the Security tab, under Users, click Select authentication type, click From the Internet, and then click Done.

This option specifies that your application will use Forms authentication, where users will log in to the application by using a login page that you will create later in this walkthrough.

1. On the Security tab, click Create User.

You are going to create two user accounts for your application.

1. In the Create User box, type information that defines a user of your application. Use the following values as guidelines. You can use any values that you like, but be sure to note your entries for use later in this walkthrough.
   * User Name Use the name, Customer01, which is the name of our first user account.
   * Password Use a password of your choice, preferably a strong password, which is a password that contains a minimum of eight characters, and includes uppercase and lowercase letters as well as punctuation.
   * E-mail Use a properly formatted e-mail address. Alternatively, you can use a non-existent e-mail address because this walkthrough does not include sending e-mail.
   * Security Question and Security Answer Type a question and answer that can be used later, if you need to recover your password.
2. Select the Active User check box, and then click Create User.
3. On the confirmation page, click Continue.
4. Repeat the preceding three steps to create one more user account named Employee01. (When you create roles for these user accounts, you will use names with all lowercase letters.)
5. Click the Back button at the bottom right corner of the page to return to the Security start page.

**Creating a User Login Page**

To test the membership and role settings that you will configure later in this walkthrough, you need to create a way for users to log in so that you can identify them.

**To create a default page for all users that displays login status**

1. Open the Default.aspx page, and then switch to Design view.

If you do not have a Default.aspx page, add one to the root of your Web site.

|  |
| --- |
| **NoteNote** |
| Be sure to name the page Default.aspx; this name is used later in this walkthrough. |

1. In the Toolbox, from the Login group, drag a LoginStatus control onto the page.

The LoginStatus control automatically takes users to the Login.aspx page, if they have not already logged in.

1. Place the cursor to the right of the LoginStatus control, press ENTER, and then type the following, including the extra space: Hello .
2. In the Toolbox, from the Login group, drag a LoginName control onto the page.

The LoginName control will display the user's name, if the user is logged in.

1. Save the page, and then close it.

You now have a home page that is available to all users. The next step is to create a simple login page.

**To create a login page**

1. Add a page named Login.aspx to the root of your Web site.
2. Open the Login.aspx page, and then switch to Design view.
3. In the Toolbox, from the Login group, drag a Login control onto the page.
4. Right-click the Login control, click Properties, and then set DestinationPageUrl to Default.aspx.
5. Save the page, and then close it.

[Establish Roles for the Web Site and Assign Users to those Roles](javascript:void(0))

Before you work with security trimming, you must set up user roles.

**Enabling Roles**

In order to make security trimming work, roles must be enabled.

**To enable roles**

1. In Visual Web Developer, on the Website menu, click ASP.NET Configuration.

The Web Site Administration Tool appears.

1. Click the Security tab, and then click Enable roles.

**Creating Roles and Assigning User Accounts to Those Roles**

Roles are now enabled, but until you create a role and assign user accounts to those roles, you cannot create access rules.

**To create roles and assign user accounts**

1. If the Web Site Administration Tool is not open, follow these steps to open it: In Visual Web Developer, on the Web site menu, click ASP.NET Configuration.

The Web Site Administration Tool appears.

1. Click the Security tab, and then click Create or Manage roles.

You are going to create two roles.

1. In the Create New Role box, type customers, and then click Add Role.
2. Repeat the preceding step to create a role named employees.
3. In the Role Name box, next to customers, click Manage.

This takes you to a page where you can select user accounts from a list to add to the customers role.

1. In the Search for Users box, click All.

This displays all of the user accounts for your Web site. You can also search by the first letter of the user's name if your list is too long.

1. Next to the user name Customer01, select the box in the User Is In Role column.

The user Customer01 is now a member of the customers role.

1. Click Back to return to the Create New Role page.
2. Repeat the preceding two steps to add the user Employee01 to the employees role.
3. Close the ASP.NET application administration page.

[Create Rules that Selectively Grant or Deny Access to Pages](javascript:void(0))

In order to test security trimming, you need to create some folders and pages for your Web site, and then set security permissions on those folders and pages.

**Creating Necessary Folders and Pages for this Walkthrough**

Your Web site will contain a site map structure that links to a few pages, which you can create now.

**To create the necessary folders and pages**

1. In Solution Explorer, right-click the root of your Web site, click New Folder, and then name the folder EmployeesOnly.

The EmployeesOnly folder will contain a page that is accessible to only some of your users.

1. Right-click the root of your Web site, click New Folder, and then name the folder Customers.

The Customers folder will also contain a page that is accessible to only some of your users.

1. In the EmployeesOnly folder, create a new page named Discounts.aspx.
2. In the Customers folder, create a new page named Support.aspx.
3. In the root of your Web site, create four new pages named Hardware.aspx, Software.aspx, Training.aspx, and Consulting.aspx.
4. In each of the new pages you have created, switch to Design view, type the page title of the top of the page, and then format the title as <H1> so that you can identify the page when you browse to it.
5. Save the new pages and then close them.

**Creating Access Rules**

The Web site that you are creating will allow users to gain access to different pages according to their roles. Therefore, you need to create some access rules that determine which roles have access to which folders.

**To set up access rules for the EmployeesOnly folder**

1. In Visual Web Developer, on the Web site menu, click ASP.NET Configuration.

The Web Site Administration Tool appears.

1. On the Security tab, click Manage access rules, and then click Add new access rules. This takes you to the Add New Access Rule page.
2. Under Select a directory for this rule, expand the root node, and then click EmployeesOnly.
3. Under Rule applies to, click Role, and then in the box, click employees.
4. Under Permission, select Allow.

The rule that you have created grants access permissions for the EmployeesOnly folder to anyone in the employees role.

1. Click OK.

The new rule is displayed in a grid in the Manage Access Rules table. When users request a page from the EmployeesOnly folder, the rules are checked in order, from top to bottom, to determine whether the user is allowed access to the page. If the user is not part of the employees role, the pages in this folder will not be displayed.

1. Click the Add new access rule link.
2. Under Rule applies to, select All Users.
3. Under Permission, select Deny, and then click OK.

The second rule for the EmployeesOnly folder makes sure that no one except users in the employees role can gain access to the folder. The rules are processed in order, from top to bottom, as you see them in the grid.

The first rule (Allow) grants access to users in the role of employees. The second rule (Deny) denies access to all other users. You can create as many Allow or Deny rules as you need for your application. When users request a page from the EmployeesOnly folder, the rules are checked in order, from top to bottom, to determine whether the user is allowed access to the page.

The next step repeats the preceding procedure for the Customers folder. However, both the customers and employees roles can be used to gain access to the Customers folder. As an example of a real-world scenario, consider a company that wants their phone support employees to have access to the same pages that customers can see.

**To set up access rules for the Customers folder**

1. Click the Add new access rule link.
2. Under Rule applies to, select the Role box, and then in the box, click customers.
3. Under Permission, select Allow, and then click OK.

The rule you have created grants access permissions for the Customers folder to anyone in the customers role.

1. Under Manage access rule, click Customers.
2. Click the Add new access rule link.
3. Under Rule applies to, select the Role box, and then in the box, click employees.
4. Under Permission, select Allow, and then click OK.

The rule that you have created grants access permissions for the Customers folder to anyone in the employees role, as well.

1. Under Manage access rule, click Customers.
2. Click the Add new access rule link.
3. Under Rule applies to, select the All Users box.
4. Under Permission, select the Deny box, and then click OK.

The first and second rules for the Customers folder makes sure that no one except users with an account in the customers role or the employees role can gain access to the Customers folder. The rules are processed in order, from top to bottom, as you see them in the grid.

The first rule (Allow) grants access to users with an account in the role of members. The second rule (Deny) denies access to all other users. You can create as many Allow or Deny rules as you need for your application. When users request a page from the Customers folder, the rules are checked in order, from top to bottom, to determine whether the user is allowed access to the page.

1. Click Done to return to the Security tab.

[Create a Site Map and Enable Security Trimming](javascript:void(0))

In this part of the walkthrough, you will create a site map to describe the navigation structure of your Web site, and then enable security trimming of site-map nodes by editing the Web.config file.

**Creating a Site Map**

To use site navigation, you need a way to describe how the pages in your site are laid out. The default method is to create an XML file that contains the site hierarchy, including the page titles and URLs.

Each page is represented as a siteMapNode element in the site map. The top-most node represents the home page, and child nodes represent pages that are deeper in the site.

**To create a site map**

1. In Solution Explorer, right-click the name of your Web site, and then click Add New Item.
2. In the Add New Item dialog box:
   1. Under Visual Studio installed templates, click Site Map.
   2. Make sure that the name in the Name box is Web.sitemap.

|  |
| --- |
| **NoteNote** |
| The file must be named Web.sitemap and must appear in the root of your Web site. |

* 1. Click Add.

1. Copy the following XML content into the Web.sitemap file, overwriting the default contents of the XML file.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_94069ba7-8757-43a6-aea0-aba7b15b3ddb');" \o "Copy to clipboard.)

<?xml version="1.0" encoding="utf-8" ?>

<siteMap>

<siteMapNode title="Home" description="Home"

url="~/default.aspx" >

<siteMapNode title="Products"

description="Our products"

roles="\*">

<siteMapNode title="Hardware"

description="Hardware we offer"

url="~/Hardware.aspx" />

<siteMapNode title="Software"

description="Software for sale"

url="~/Software.aspx" />

<siteMapNode title="Discounts"

description="Employee Discounts"

url="~/EmployeesOnly/Discounts.aspx" />

</siteMapNode>

<siteMapNode title="Services"

description="Services we offer"

roles="\*">

<siteMapNode title="Training"

description="Training"

url="~/Training.aspx" />

<siteMapNode title="Consulting"

description="Consulting"

url="~/Consulting.aspx" />

<siteMapNode title="Support"

description="Support"

url="~/Customers/Support.aspx" />

</siteMapNode>

</siteMapNode>

</siteMap>

The Web.sitemap file contains a set of siteMapNode elements that are nested to three levels. The structure of each element is the same; the only difference among them is their location within the XML hierarchy.

The URL of the pages that are defined in the sample XML file are unqualified, meaning that all of the pages will be treated as having URLs relative to the application root. However, you can specify any URL for a given page — the logical structure that you define in the site map does not have to correspond to the physical layout of the pages in folders.

The roles attribute is only set on two nodes in the site map: Products and Services. These two nodes do not have a url attribute. ASP.NET uses the access rules of an .aspx page to verify whether the site-map node should be visible to a user. Without a url attribute, no user can view the page if security trimming is enabled. Setting the roles attribute to an asterisk (\*), or a wildcard character, expands access to everyone.

1. Save the file, and then close it.

**Enabling Security Trimming**

The site map that you created uses the default site-map provider, XmlSiteMapProvider. For this reason, the Web.config file does not need to contain a [siteMap](http://msdn.microsoft.com/en-us/library/1e333zt4(v=vs.100).aspx) element because ASP.NET has an intrinsic configuration for default site maps. However, security trimming is not enabled by default, so the [siteMap](http://msdn.microsoft.com/en-us/library/1e333zt4(v=vs.100).aspx) element must be set.

**To enable security trimming**

1. In Visual Web Developer, right-click the root of your Web site, and then click Refresh Folder.

This makes the Web.config file appear in Solution Explorer. The Web.config file can be created by selecting Add New Item or by using the ASP.NET Configuration tool as we did earlier in this walkthrough.

1. Open the Web.config file.
2. Look for a [siteMap](http://msdn.microsoft.com/en-us/library/1e333zt4(v=vs.100).aspx) element in case it already exists.

If the [siteMap](http://msdn.microsoft.com/en-us/library/1e333zt4(v=vs.100).aspx) element does not exist, paste the following code inside the [system.web](http://msdn.microsoft.com/en-us/library/dayb112d(v=vs.100).aspx) element, being careful not to paste it inside any other element.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_4160df6a-7b0c-4b65-bde7-e67b4cd7f6c1');)

<siteMap defaultProvider="XmlSiteMapProvider" enabled="true">

<providers>

<add name="XmlSiteMapProvider"

description="Default SiteMap provider."

type="System.Web.XmlSiteMapProvider"

siteMapFile="Web.sitemap"

securityTrimmingEnabled="true" />

</providers>

</siteMap>

1. Save the Web.config file, and then close it.

The security-trimming feature uses URL authorization on each request to determine whether a user has access to a URL that is associated with a siteMapNode element. This extra work reduces performance depending on the number of nodes being authorized. When security trimming is enabled, you can use the following methods to improve performance:

* Limit the number of nodes in the site-map file Site-map files with more than 150 nodes can take substantially longer to perform security-trimming operations.
* Set the roles attribute explicitly on siteMapNode elements Note that setting the roles attribute to an asterisk (\*) should be used for nodes that can safely be displayed to any client. The presence of a roles attribute allows ASP.NET to bypass URL authorization for the URL that is associated with the siteMapNode when a user belongs to one of the roles listed in the attribute.

[Create a Navigation Menu on a Page](javascript:void(0))

Now that you have a site map and some pages, you can add navigation to your site.

Alternatively, you can use a SiteMapPath or Menu control to display site navigation. For more information, see [Walkthrough: Adding Site Navigation to a Web Site](http://msdn.microsoft.com/en-us/library/w6ws38fw(v=vs.100).aspx).

**Creating a Navigation Menu with the TreeView Control**

You will use the TreeView control to act as a collapsible navigation menu.

**To add a navigation menu**

1. Open the Default.aspx page, and then switch to Design view.
2. In the Toolbox, from the Data group, drag a SiteMapDataSource control onto the page.

In its default configuration, the SiteMapDataSource control gets its information from the Web.sitemap file that you created previously, so you do not need to specify any additional information for the control.

1. In the Toolbox, from the Navigation group, drag a TreeView control onto the page.

The TreeView Tasks menu appears.

1. On the TreeView Tasks menu, in the Choose Data Source box, click SiteMapDataSource1.
2. Save the page.

**Creating a Navigation Menu with the SiteMapPath Control**

You will use the SiteMapPath control to act as a breadcrumb-style navigation display that shows users the path back to the root page.

**To add a navigation menu**

1. Open the Hardware.aspx page, and then switch to Design view.

To illustrate how the SiteMapPath control works, place the control on a deeply nested page.

1. If you added a title inside <H1> tags on this page, place your cursor at the right side and press ENTER.
2. In the Toolbox, from the Navigation group, drag a SiteMapPath control onto the page.

The SiteMapPath Tasks menu appears, but there is no need to configure a data source.

1. Save the page.

[Testing Security Trimming](javascript:void(0))

Your site is now ready for testing.

**To test security trimming**

1. Switch to the Default.aspx page, and then press CTRL+F5 to run the page.

In the TreeView control, you should not see the Discounts or Support links.

1. Click Login.
2. Log in as Customer01, which is a user account that is in the customers role.

In the TreeView control, you should now see the Support link.

1. Click Logout, and then click Login.
2. Log in as Employee01, which is a user account that is in the employees role.

In the TreeView control, you should now see both the Discounts link and the Support link. As an employee, you should see all of the available links because none of the links are restricted for employees.

1. On the Default.aspx page, click Hardware so that you can see the SiteMapPath control.

The SiteMapPath control should display the path from the home page to the Hardware page.

In the current state of the Web site, the navigation tree appears only on the home page. You could add the same SiteMapDataSource and TreeView controls to each page in the application to display a navigation menu on each page. Alternatively, you could put the navigation tree on a master page so that it automatically appears with every page that is referenced by the master page file.