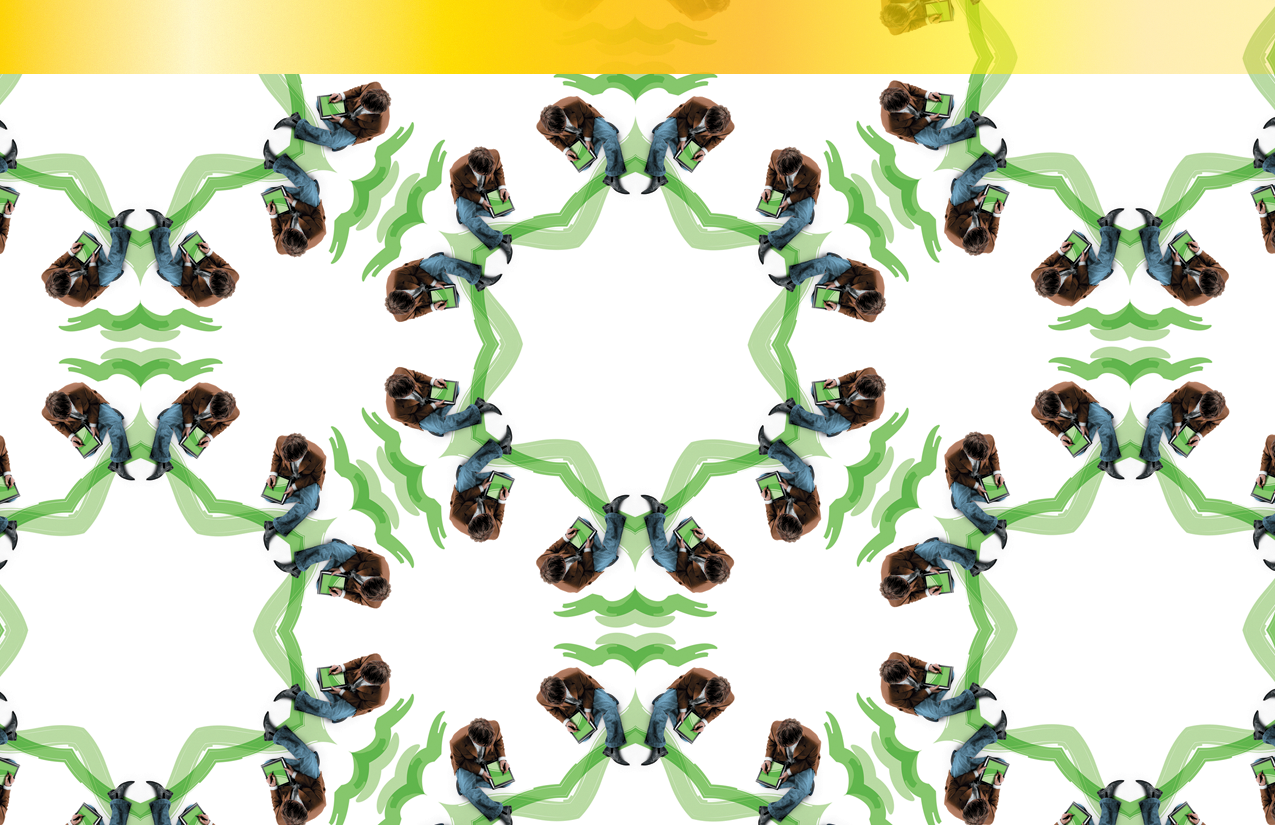
|  |
| --- |
|  |

Microsoft® Expression® Web

Reviewer’s Guide

[Introduction 3](#_Toc150913276)

[Introducing Expression Web 3](#_Toc150913277)

[Expression Web Customer Profile 4](#_Toc150913278)

[Expression Web Key Messages 6](#_Toc150913279)

[Overview of Feature Areas 7](#_Toc150913280)

[Standards-Based Web Sites 7](#_Toc150913281)

[Sophisticated CSS-Based Layout & Formatting 7](#_Toc150913282)

[Rich Data Presentation 8](#_Toc150913283)

[Powerful Server Technology 8](#_Toc150913284)

[Top 20 Reasons to buy Expression Web 10](#_Toc150913285)

[Hands On Exercises 13](#_Toc150913286)

[About this section 13](#_Toc150913287)

[Product and Company Facts 35](#_Toc150913288)

[PR Contacts 35](#_Toc150913289)

[System Requirements 35](#_Toc150913290)

[Company Headquarters 36](#_Toc150913291)

# Introduction

Welcome to the reviewers’ guide for ***Microsoft® Expression® Web***. The purpose of this document is to assist your evaluation of the product.

The Microsoft Expression family of products are being introduced by Microsoft to address the needs of the professional design community who wish to design and create content based on the Microsoft technology stack.

Whether that is for the desktop or the web, the Expression products will allow designers to fully engage with developers to finally bridge the gap in the designer/developer workflow. This will bring together the best minds in design and development to create applications and sites with a focus on great *User Experience* (UX).

It’s a subject that Microsoft is passionate about and Windows Vista and the 2007 Microsoft Office system are both examples of what can be achieved in this space. The combination of the Expression tools for designers with Visual Studio 2005 for developers will see the birth of a new era in software and website development with UX at its center.

# Introducing Expression Web

***Microsoft Expression Web*** is the first product to be released in the Microsoft Expression family, a series of new professional design tools. The product is based in part on technology used by FrontPage 2003 which will enable most legacy sites to be edited by Expression Web, but a new design rendering surface has been created that accurately renders code based on today’s standard XHTML.

Expression Web will bring excitement to the professional web design community with its focus on building standards based sites using all of today’s recommended technologies such as XHTML, CSS, XML and XSLT.

A fully featured professional design surface lets the user be visually creative in the knowledge that the underlying code being created is clean and purposeful.

Couple that, with the ability to design web applications using ASP.NET 2.0 and you have in Expression Web a new and exciting tool for now and the future.

# Expression Web Customer Profile

Expression Web will be used by customers worldwide for a wide range of activities within the web design process. These customers will focus on areas of web design that fall within these broad categories.

* Website UI
* Web Application Development
* CSS
* Accessibility
* Internationalization

### Website UI

These users will enjoy the standards based approach to design that Expression Web offers. The design surface allows them to create sites with the confidence that the underlying code written by the application fully adheres to modern W3C standards such as XHTML, CSS, XML and XSLT.

### Web Application Development

Creating web applications can be done with a range of technologies, all of which require the designer to deal with complex code entities that belong in the world of the developer. Expression Web, by focusing on the power of ASP.NET 2.0 allows designers to drag and drop a wide variety of technology centered controls onto their design surface and then style them without having to touch the code view. The project can move back and forward between coders and designers as often as needed, with developers using the power of Visual Web Developer within Visual Studio 2005 and designers using Expression Web.

### Cascading Style Sheets

Separating design from content is one of the primary goals of CSS and the creation of designs using CSS is fully supported within Expression Web. Indeed the outdated practice of designing with tables and font tags is virtually eliminated from the design surface. Design decisions taken with the visual tools are written as CSS according to the version chosen by the designer. The resulting rules can then be consolidated and managed using a sophisticated drag and drop panel.

### Accessibility

A wide range of users specifically focus on the workflows needed to build websites that fully conform to accessibility. It’s an aim that everyone should have when designing sites, so Expression Web plays its part in helping with that. By building sites that conform to the W3C standards, it already takes users part of that way. The program also prompts for required information where appropriate. Finally, users can run an exacting set of reports on page and site content, the results of which hyperlink through to the various standards sites to provide information on areas of non-conformity.

### Internationalization

The need to build websites that work across languages and cultures has never been greater and users need programs that fully support the UTF-8 (Unicode) character set as well as dealing with issues like non-left to right text sets. It will often be the case that design teams will incorporate users with skills in multiple languages, so having one tool that successfully supports this will be key to their success.

# Expression Web Key Messages

### New Era, New Tool

Reduce complexity and ease data integration by using powerful design tools and task panes to quickly incorporate XML data. Seamlessly integrate Web design and development teams with Expression Web and Visual Studios’ superior support for XML, ASP.NET and XHTML.

### Beautiful Inside and Out

Unleash your creative ideas and bring your Web sites to life with sophisticated CSS design features. Visual designers, specialized task panes, and tool bars give you precise control of page layout and formatting.

### Passionate about Standards

Build dynamic, interactive pages that harness the power of the Web to deliver superior quality. Built-in support for today’s modern Web standards makes it easy to optimize your sites for accessibility and cross-browser compatibility

# Overview of Feature Areas

## Standards-Based Web Sites

Create CSS-based, XHTML 1.0 Transitional–conforming Web sites by default, and work better across browsers, simplifying deployment and maintenance. Configure flexible schema settings to support all combinations of HTML/XHTML/Strict/Transitional/Frameset and CSS 1.0/2.0/2.1 in addition to browser-specific schemas. Validate your site with compatibility and accessibility reporting and against Section 508 and W3C Content Accessibility Guidelines (WCAG).

**XHTML compliance** - Design sites to today’s exacting standards. It’s easy with Web Designer’s deep support for XHTML..

**Full schema support** - Choose any schema you want to adopt—Web Designer will automatically build appropriate code, and the compatibility checker highlights any non-conforming elements.

**Real-time standards validation** - Get instant feedback on standards violations. This means corrections can be made as part during the design process, whether in design or code view.

**Accessibility checking** - Use the built-in accessibility checker to make sure your Web sites can fully meet screen reader accessibility standards, with support for WCAG and Section 508.

## Sophisticated CSS-Based Layout & Formatting

Generate elegant, modern CSS layouts with state-of-the-art surface design tools and the ability to directly manipulate positioning, sizing, margins and padding. A sophisticated CSS rendering engine inside Expression Web gives you the confidence to make design decisions on the fly, and drag-and-drop style management lets you try out CSS styles on the page and then migrate them to a central repository. Typing aids and IntelliSense® in the code view make your process efficient and fast.

**Professional design surface** - Design high-quality Web sites and applications with a state-of-the-art, professional toolset and design surface. Its powerful and easy-to-use tools let you directly manipulate margins and padding throughout the document.

**Intelligent CSS style creation** - Focus on the design process and let Expression Web take care of CSS style creation, saving you from repetitive work throughout your site.

**High-quality CSS rendering engine** - Make design decisions with confidence. You’ll see a faithful representation of the final browser-rendered page with a powerful, built-in CSS rendering engine.

**Drag-and-drop style management** - Try out designs with local in-page CSS styles and then easily migrate styles to a central repository using

**Standard or Quirks rendering** - Render pages in Standard or Quirks mode depending on the doctype setting of the page for maximum browser compatibility.

**Intelligent style renaming** - Let Expression Web’s metadata keep track of style usage, allowing to you rename styles instantly. No “find and replace” needed.

**Site and page templates** - Get up to speed fast with professionally designed templates that are fully XHTML and CSS compliant.

**CSS reporting** - View detailed reports on CSS style usage across sites in the Results pane to keep track of where you’ve used styled pages.

**Tag property grid** - Retrieve and set all valid properties for a selected tag. Access all the power of the HTML language through a single palette.

**CSS design time preferences** - Fine-tune how CSS code is generated throughout an application. You have complete control over the details of Expression Web’s code generation.

## Rich Data Presentation

Build and format views of industry-standard XML data using drag-and drop tools for quick visualization of formatting that is consistent with the rest of your site. An XPath Expression Builder allows you to create complex queries, and XSL Transform (XSLT) support gives you the flexibility to present data any way you want. XML data can also be retrieved and presented via live RSS feeds that match your site’s formatting.

**Customized views of XML data** - Design customized views of industry-standard XML data using powerful drag-and-drop tools, and see how your data will look within your CSS site design.

**XSL Transform support (XSLT)** - Present data using XSLT, even editing views on live XML data. Designers get the flexibility to represent XML data any way they choose.

**Powerful data manipulation** - Fine-tune the raw data presented on the page with tools to sort, filter, and apply condition formatting.

**RSS Feeds** - Retrieve and present XML data from live RSS feeds on your site, styled to match your own site’s look and feel.

## Powerful Server Technology

Harness the power of ASP.NET 2. to transform your sites into dynamic, interactive Web applications, with integrated support for server and user controls and no code data binding required. The controls toolbox, properties grid, and on-control “action menus” in Expression Web make inserting and configuring ASP.NET controls a snap, while ASP.NET Master Pages make site-wide changes quick and easy.

**ASP.NET pages** - Create, edit, and deploy Web pages with deep support for the newly released ASP.NET 2.0 standard. Capture the power of ASP.NET with easy drag-and-drop controls.

**Master Pages** - Maintain the correct look of your site with the use of server-rendered ASP.NET master pages to speed up the design process and allow easy site-wide changes. Nested master pages keep you on top of complex designs.

**ASP.NET controls** - Access ASP.NET controls and their associated properties with the same power as Microsoft® Visual Studio® 2005, but rendered within the design surface of Expression Web.

**ASP.NET Development Server** - Preview ASP.NET pages locally, without the need for a full Web server. With the ASP.NET Development Server, you can make design decisions with full reference to the page’s data.

**No code data binding** - Use a drag-and-drop interface that accesses the power of ASP.NET with no coding requirement to visually bind data sources to controls on the design surface.

# Top 20 Reasons to buy Expression Web

### 1. Professional Design UI

Building a modern website requires a program that places the tools you need where you need them. With its array of task panes and common task menus strategically placed where they are needed, you can concentrate more on the process of design and less on searching thorough complex menu structures.

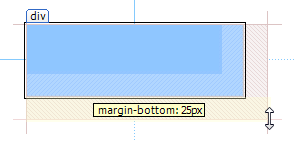
### 2. Standards Based Site Design

Expression Web builds sites according to today’s modern standards including XHTML, CSS, XML and XSLT. The benefits to the web designer are far reaching, from better rendering in browsers to more relevant indexing in search engines. By helping designers do things right from the outset, you can be more productive.

### 3. CSS and Site Templates

Kick start your designs with a range of professionally designed, standards compliant, full site templates from a broad range of categories. Or start your CSS design from a range of DIV based layouts.

### 4. CSS Layout

Today’s sites are built using Cascading Style Sheets to separate design from content. It’s a complex technology made easier with the design focussed tools in Expression Web, putting the user in complete control. From visual drag and drop control of margins and padding through to a visual hierarchy of style application, it’s easy to take control of your site’s design.

### 5. CSS Style Application

Choose between having your CSS styles created automatically or switch to manual control for fine grain control over the type of rule created and its precise location - the choice is yours.

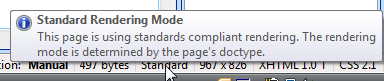
### 6. CSS Management

Expression Web has the leading set of tools to manage where CSS rules are created, where they’re located and the order in which they’re applied. All achieved visually without needing to view the underlying code. It’s never been easier to manage complex CSS in a site.

### 7. Property Task Pane

Quickly access the full range of CSS and tag attributes using either of the property task panes. You can reorder the list to find the ones you need and view just a summary of “in use” CSS properties for fast reference.

### 8. Page Rendering

Accurate rendering of a page in the design environment is essential to allow the designer freedom to create. Expression Web has what we believe to be the most accurate rendering engine available today, switching from standard to quirks rendering depending on the doctype in use. Have confidence that your designs will look as good in the browser as they do in your design tool.

### 9. Rich Data Presentation

Working visually with XML data has never been easier. Simply drag and drop an XML file and it’s immediately rendered by an XSL stylesheet that you can reformat to match your site’s style. Now you can include, filter and sort and style data from any XML source easily.

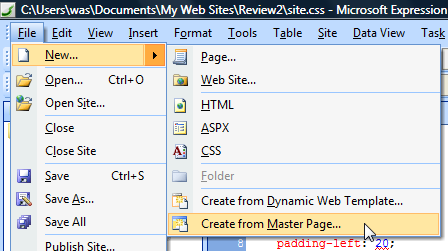
### 10. ASP.NET 2.0

Bring your sites to life with the latest version of ASP.NET offering a wide variety of controls from navigation to calendars, login functionality to database integration. All rendered in the design surface without the need to resort to coding.

### 11. Control Common Task Popout

Easily find the most common control configuration tasks from the popout menu. Access a range of tasks from sorting and filtering data, to formatting the look and feel of controls.

### 12. Master Pages

Maintain consistent design throughout your ASP.NET site by basing page designs on a Master Page that is always current. The only time the page is built is when it’s served by the application server. Nest Master Pages to form complex design hierarchies.

### 13. Code IntelliSense

When it’s time to edit the page’s code, rely of the code complete features of IntelliSense to accurately offer you a choice based on your chosen doctype and CSS version.

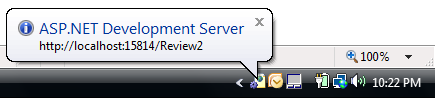
### 14. Code Style Hyper-linking

Easily find the source of your applied style in code view by control clicking on any id or class name to be taken straight to and select the relevant rule.

### 15. Code Snippets

Reuse chunks of your favorite code by saving them as code snippets and quickly apply them where needs with a few quick keystrokes. An aid to productivity and accuracy.

### 16. ASP.NET Development Server

View ASP.NET pages on your design workstation without the need for any complex web server. The ASP.NET development server launches when it’s needed to serve your ASP.NET pages without taking up the system resources of a full web server.

### 17. Reporting

A full suite of reporting across a broad range of technologies means you can be confident that your site will be deployed error free. Check CSS usage and errors, accessibility against Section 508 and WCAG and compatibility with alternative schemas and browsers.

### 18. Visual Studio Integration

Work as part of a team with confidence that your sites created with Expression Web are fully compatible with Visual Studio to allow further development of your site’s ASP.NET functionality.

### 19. HTML Optimization

As you deploy your site to a Web Server, use the powerful optimization options to remove commenting and white space to ensure the files are ultimately downloaded and served as swiftly as possible without affecting your local copies.

### 20. Extensibility

Add to the native functionality of Expression Web with a range of Microsoft and third party add-ins or create them yourself with reference to the program’s SDK.

# Hands On Exercises

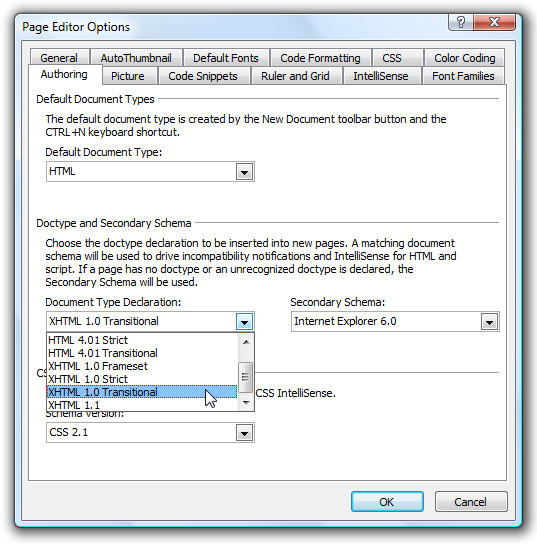
## About this section

Accompanying this reviewer’s guide is a PDF of the printed Quick Start Guide that ships with Expression Web along with the 2 hour getting started tutorial video. We realize however that not all reviewers have time to complete full tutorials on products they are reviewing, so as an alternative to these resources, we are proving some simple hands on exercises as an alternative to these tutorials to enable you to get into the product features quicker. There are 20 exercises that map to the top 20 features however the order is different to allow a better flow to the exercises. We do however encourage you to use the information contained within the Quick Start Guide to accompany these exercises.

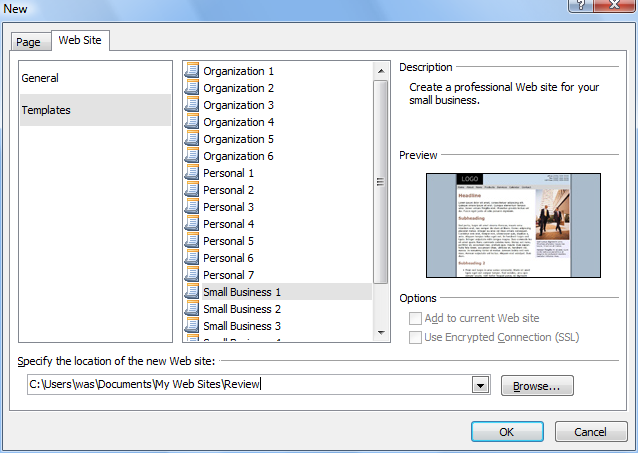
### 1. Professional Design UI

* Install and open Expression Web.
* By default, the six most commonly used task panes are visible in 4 windows
  + Folder list
  + Properties
    - Tag Properties
    - CSS Properties
  + Toolbox
  + CSS
    - Apply CSS
    - Manage CSS
* You can move the taskpanes around by dragging and dropping them throughout the UI.
* Open additional task panes from the Taskpane menu.
* The current window layout is remembered.
* Reset the taskpanes to standard by choosing **Task Panes > Reset Workspace Layout.**
* Once a page is opened, you have the option of viewing the page in Design view, Code view or a combination view, Split view.

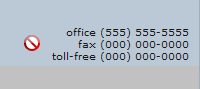
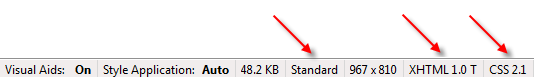
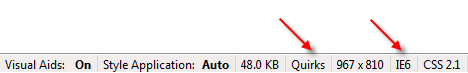
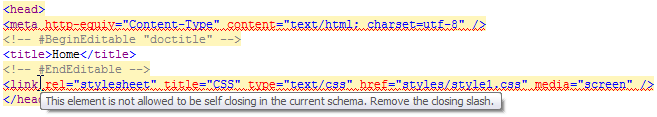
### 2. Standards Based Site Design

* The user is given many options to define how they want to work with Expression Web. Initially, a decision has to be made as to which standard you wish to build to. This affects all new sites created from scratch. The templates all use XHTML 1.0 Transitional and CSS 2.1
* Choose **Tools > Page Editor Options…** and then the Authoring tab  
  
* The lower half determines the doctype you wish to use which defaults to XHTML 1.0 Transitional.
* The Secondary Schema comes into play with legacy sites where there is no doctype, or the doctype is malformed. This allows you to force the notification system and Intellisense in code view to work to your desired standard.
* Choose **Cancel** to return.

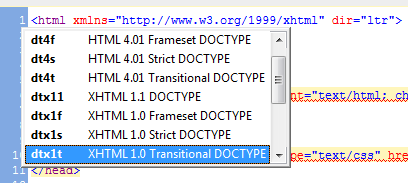
### 3. CSS and Site Templates

* Ensure there is no site open by choosing **File > Close Site** if enabled.
* To create a new page, choose **File > New… > Page…**
* Explore all the single page options, in particular the CSS layouts.
* Now flip views to the website creation tab at the top of the dialog.
* Choose the template subsection and explore the range of site templates. These are built using Dynamic Web Templates so they are not dependant on ASP.NET
* Create a site based on “Small Business 1” by selecting it in the list, and specifying a location where you wish the site to be saved on your hard drive.  
  
* You can now see in the folder list, a fully prepared website which the user can modify to their needs by modifying the graphics, colors and default text.
* Open the site’s template file, **master.dwt** from the folder list.
* This is the document from which all the other pages in the site are built.
* Change one of the phone numbers in the top right hand side to read something different like (555) 555 5555. As soon as you make a change, the file name in the file’s tab has an asterisk added to denote the change.
* Now choose **File > Save** and you are alerted to the fact that 15 files are linked to this template. Click **Yes** to update all 15.

### 4. Page Rendering

* Expression Web has 2 rendering modes in design view, a high quality standards based mode and a legacy “quirks” mode based on IE5. The standards mode is not based on any installed browser, it is an independently built technology within the program. To explore these 2 modes, first open the site’s home page **default.htm**.
* Note 2 things. Firstly how the top of the document is locked with a symbol to show the areas that are locked by the template. Secondly note your new phone number appeared in this document.  
  
* Note in the status bar, the standards mode, the doctype in use and CSS version.  
  
* Take a good look at the page layout and then choose Code view from the tab at the bottom of the document.
* Highlight the first line of code, starting with <!DOCTYPE…> and delete it, so the first line of code now becomes <HTML.
* Now switch back to Design view and you should notice a change in the way the page looks, and now the status bar has change to show that we are now in Quirks mode and the secondary browser schema IE6 is now in place.  
  
* This gives us a chance to look at the schema error feature, because now Expression Web has a perfectly good XHTML document but without the required doc type. Change back to code view, and note the red underlined places in the code. Hover over any of them and read the tooltip explaining the error and in many cases, how to fix it.  
  

### 5. Code Snippets

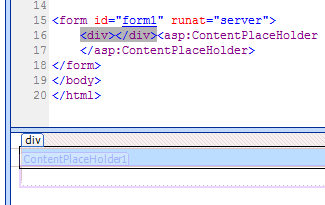
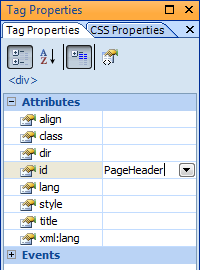
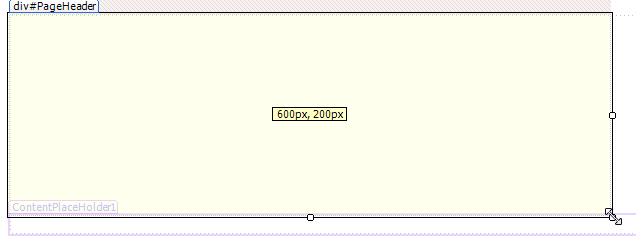
* To repair the broken doctype, place the cursor before the first character of line one, and press **Ctrl-Enter** to bring up the code snippet feature. Use the arrow keys to scroll down the list to the XHTML 1.0 Transitional doctype entry and press enter to add to the document.  
  
* Additional snippets can be added by the user by choosing **Tools > Page Editor Options…** and choosing the **Page Snippets** tab.

The next section is dealing with the Cascading Style Sheet (CSS) technology and exploring the ways of dealing with the management of this feature. We will close down the previous site and start a new site based on the ASP.NET Master Page feature.

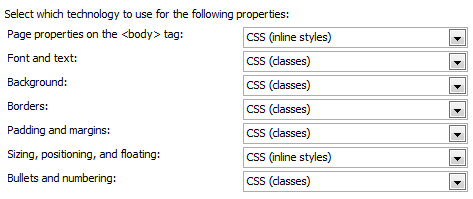
### 6. Master Pages

* Close the existing site which can be deleted from your hard drive using the File Explorer. **File > Close Site**.
* We’ll start by creating a new website using **File > New… > Web Site…**
* From the **General** section choose **Empty Web Site**, select a location for your site, calling it ReviewCSS and then click **OK**.
* You now have an empty site into which we need to create our ASP.NET master page.
* Choose **File > New… > Page…** and from the **ASP.NET** section choose **Master Page**. (Note the option options available to you, including at the bottom right, the selection of language for the code behind file). Click **OK**.
* Immediately save the page as “Site.master”.
* You can see on the page, there is one region called Form1. Click inside it and you will find yourself in something called a ContentPlaceHolder. This is a special region of a master page into which custom content can be created on any page made from this master. You therefore design the look and your feel of the page as desired and determine which parts of the page can be “unlocked” for editing by making them ContentPlaceHolders.

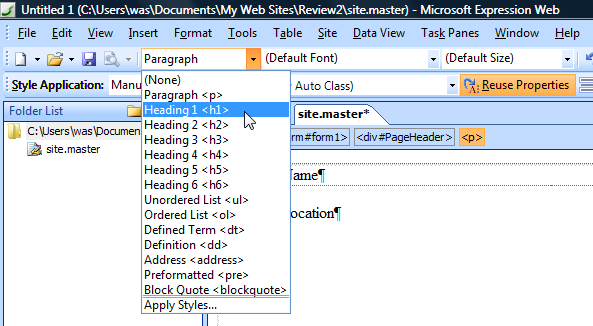
### 7. CSS Layout

* Modern sites that fully conform to the recommended notion of separating design from content are designed and built using CSS-P, with DIV tags and layers (DIV tags with absolute positioning). This is a change from the previous often used practice of using tables.
* You can create a DIV tag in many ways, you can type it in code view, you can add it from the Toolbox or you can use the Layers taskpane to create a layer. Because it matters where in the tag hierarchy these elements are place, it’s a good idea to start to position these basic elements in Split view. This ensures that you have the element placed where you want it precisely.
* Choose Split View from the bottom of the page.
* We want to place a DIV tag before the default ContentPlaceHolder to hold a page header, so ensure that the cursor is in front of this element in the code and then double click on the <DIV> tag in the Toolbox.
* Move down into the Design part, and click on the newly created DIV to activate it (click in the space above the existing ContentPlaceHolder). You should end up with a view like this.  
  
* It’s a good idea to give each DIV on the page a unique ID, so in the Tag Properties taskpane (lower left), in the ID field type “PageHeader”  
  
* We can now resize this DIV to be whatever size we need, by grabbing the lower right handle. Resize it to roughly 600 pixels wide and 200 pixels deep. Precision doesn’t matter at this point.  
  

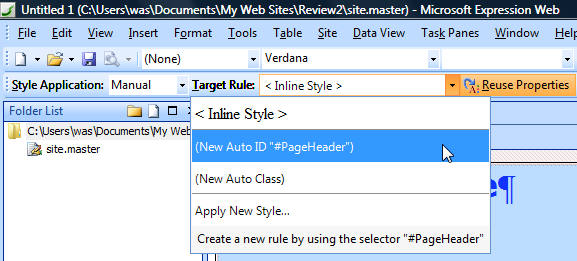
### 8. CSS Style Application

* If you refer back to the code section, you will notice that the size of the DIV has been applied as an inline style:  
  <div id="PageHeader" style="width: 600px; height: 200px">
* Expression Web is currently running in a mode where is makes its CSS decisions based on a set of inbuilt rules. You can see these rules by choosing **Tools > Page Editor Options** and then the **CSS** tab.
* Note in the section “Sizing, positioning and floating” it is set to create CSS (inline styles). You can change this to create classes if you wish.  
  
* If you want to have complete control over each individual decision as it is made, you can set to program to Manual mode. In the dialog box, choose **Manual Style Application** and then **OK**.
* A new toolbar appears, the Style Application toolbar, which can be docked near the top of the screen underneath the Common toolbar if desired.
* In this mode, you can use the Target Rule popup to indicate how you wish to treat the next design choice.
* The “Reuse Property” button on the toolbar instructs Expression Web to climb up the rule cascade to find properties. This bears some additional explanation. If you make a change to a typeface on a selected paragraph and that typeface is actually being set in a rule applied to the containing DIV, the Reuse Property feature will stop a new rule being created and instead make the change to the class attached to the DIV tag.
* Using the CSS Style Application feature, the user is in complete control over how Expression Web creates rules.

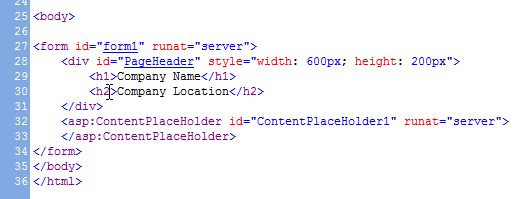
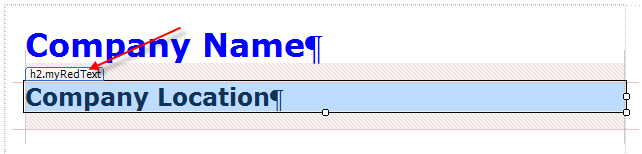
### 9. CSS Management

* Let’s create some text and styles to see how easy it is to manage complex rule hierarchies with Expression Web.
* Place the cursor into the DIV you created and type 2 lines of type.  
  Company Name  
  Company Location
* You can select the content tag for selected text from the Common toolbar
* Click in the first line and assign it to be an H1 tag from the popup  
  
* Click in the second line and assign it to be an H2 tag in the same way.
* Let’s make all the text on our page use the Verdana typeface. A good way to do this is to create a rule attached to the body element. In the Apply Style taskpane in the lower right, click the **New Style…** button
* This is the dialog where you create or edit all the properties of a rule in one place. Take a moment to explore all the categories.
* From the Selector popup at the top right, choose “body”. Then from the font category choose the font-family “Verdana” (you can type a v in the box to speed up selection). Click **OK** and both lines of type change.
* Now let’s make a rule to change the first line to blue. We could create a class, or we could create an H1 element rule. Let’s do the latter.
* Again, click the **New Style…** button as before, choose h1 in the Selector popup and this time choose Blue from the color.
* I now want to move the type down from the top and in from the left of the box. I can do this by adding padding to the box of the DIV tag. But where should this rule be applied? Again, although there are various ways, none are right and wrong. Let’s create a rule to be applied to the ID we assigned to the DIV earlier – we called it PageHeader if you recall? We’ll do this using a new technique.

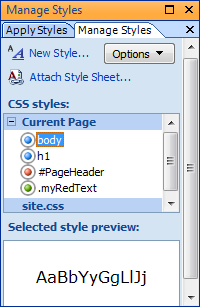
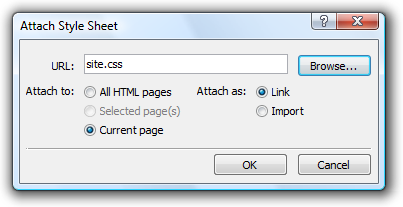
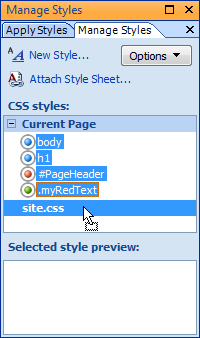
### 10. Visual Margins and Padding

* Click first in the type and then use the tag breadcrumb trail to select the containing DIV  
  
* We want to be sure that any changes are applied to a rule attached to the DIV’s ID “PageHeader” so return to the Style Application toolbar and from the Target Rule popup choose **New Auto ID “#PageHolder”**   
    
  It gives you this choice because it detects that you are within a tag that has had an ID applied.
* Now carefully place the cursor on the horizontal hairline that has appeared at the top right of the DIV. When you hover over it, a tooltip saying “margin-top: (15px)” should appear. If you were to click and drag down now, you would increase the margin above the DIV. We want to change the padding inside the DIV however, so hold down the **SHIFT** key before clicking and dragging down – this will change the padding. Apply a padding of 20px.
* Now do the same from the bottom left, not forgetting the **SHIFT** key again  
    
  Once you have finished, you can see that with the DIV still selected, you get a colored visual cue of both the padding inside (blue) and the margin above (pink) the DIV. This feature makes it easier for designers to precisely control their page elements and have full visibility into where margins and padding are affecting their designs.
* Let’s take a chance to make some more CSS changes using the code view.

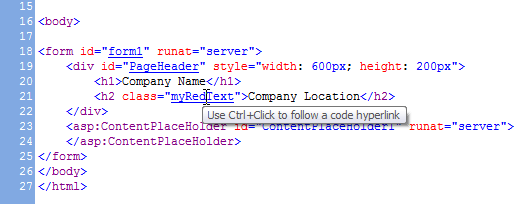
### 11. Code IntelliSense

* Sometimes it makes sense to work in code view, even for a designer. It’s often quicker and more accurate to make the change to the code. Let’s assign a class this time to our second line of text, “Company Location”
* Change to code view and carefully place the cursor just before the closing > of the H2 tag  
  
* Type a **SPACE** and immediately you get a popup menu of things you might be thinking of typing. We want to add a class attribute, so type a **c** and the list skips straight to class. Just hit **ENTER** to add the rest automatically and because the class property needs to have an attribute between quotes, they are provided and the cursor is place in between them. Now type **myRedText** to create a new class name.
* Quickly flip back to design view and see that this change has been applied to the selected line of text  
  
* Go back to code view and find the style block that Expression Web has created at the top of the document. You now need to place the cursor after the last rule in the block which should be the ID “#PageHeader”, immediately in front of the </STYLE> tag and then press **ENTER** to create a blank line. We’re creating a class this time and classes begin with a period “.”. Type the period now, and a popup should appear with the name of the class we used earlier. IntelliSense remembered that we used the class earlier and now “reminds” us to create it. **ARROW DOWN** to the name and press **ENTER**.
* Type another **ENTER** and then a **{**
* The next popup gives the full range of CSS properties available. Typing **co** will find **color** in the list. Type **ENTER** to select it. You are then prompted for the actual color. **ARROW DOWN** to **Red**. Finally, close the statement with a **;** (semi colon).
* Type another **ENTER** to add a line and then type **fo** to make font appear in the list, **ARROW DOWN** to enter “font-family” and from the list choose the family starting with **Ariel**.

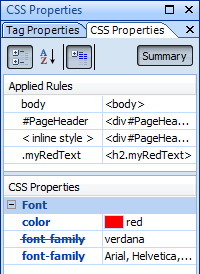
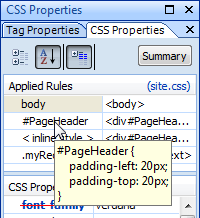
Now we have a range of rules in our master page, it makes sense to now consolidate these into an external style-sheet document. This keeps our master page neat and makes these rules accessible to other pages if required.

* Change back to design view and activate the Manage Styles taskpane in the lower right. It should contain our list of 4 styles.  
  
* Note the color coding of the list, blue for elements, red for IDs with classes appearing in green. The circle around the dots indicates that the style is in use within the current document.
* To create our external CSS document choose **File > New… > CSS**
* Choose **File > Save** and name the document “site.css”
* Return to the “site.master” document and then from the Manage Styles task pane, click the **Attach Style Sheet…** button and then browse to and select this newly created document.  
  
* To move all the styles into this document, shift-select the 4 styles in the Manage Styles task pane and then drag them onto the site.css filename.  
  
* To check this has happened, go to the code view of site.master and you should see the style block is now empty (not removed, it might be needed again). You now have one extra choice to make in the Style Application toolbar – *where* the rule is created.

### 12. Code Style Hyper-linking

* During the site creation process, you may end up with many hundreds of rules within your site and navigation can be become problematic.
* Whilst still in code view of site.master, note how all the styles used have blue hyperlinks.  
  
* **Ctrl-click** on any of these to be taken immediately to the rule within any document, internal or external.

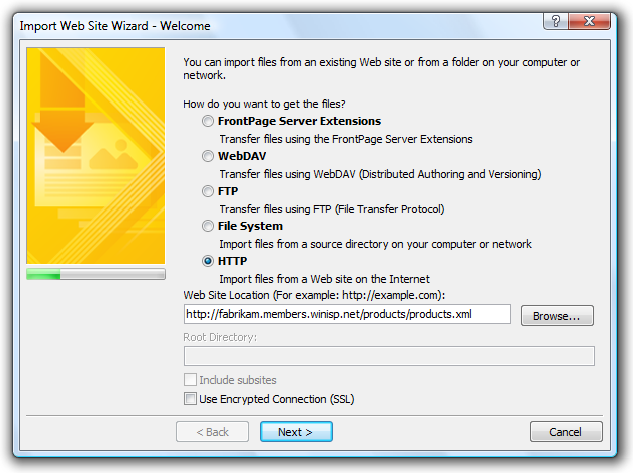
### 13. Property Task Pane

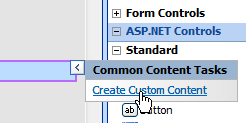
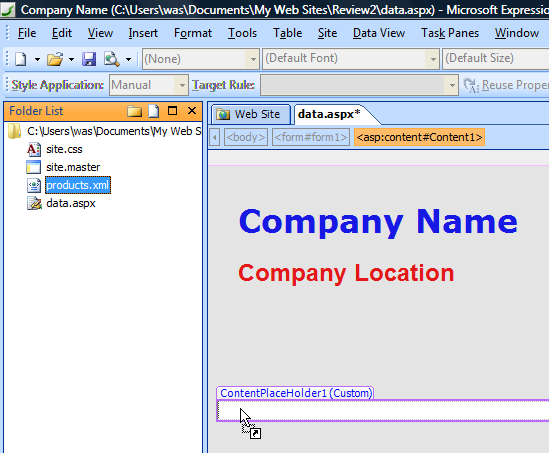
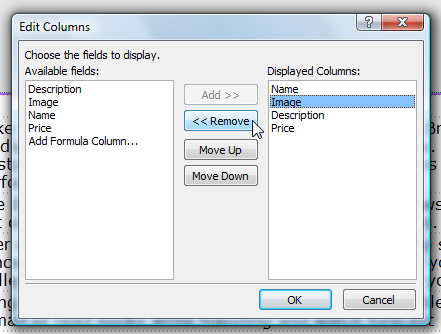
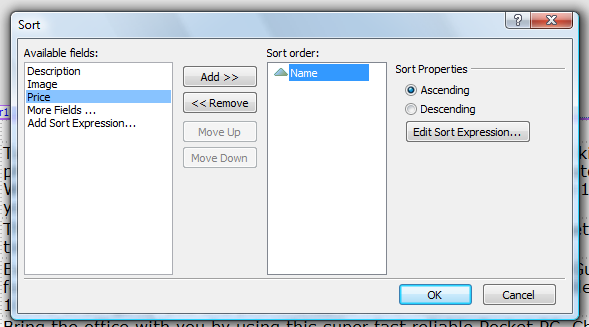
* Finally, we will look at the 2 property task panes and how they can be used to explore the attributes set within the document.
* Click to activate the H2 text “Company Location” making sure it is completely selected using the tag breadcrumb trail if necessary.
* Click to activate the Tag Property task pane if it is not already selected.
* You can now see in the Tag Property task pane that the tag has a class of “myRedText” applied, as you would expect.
* If you now click back down the breadcrumb trail on the DIV, you can see that the Tag Property task pane shows the DIV’s ID “PageHeader” along with the inline style. Remember of course, that attributes can also be applied and edited in this task pane.
* Now select the CSS Property task pane and assuming the DIV tag is still selected, you should see the three rules in the cascade that apply to that tag, you can also think of them as being three rules that are having an effect on that tag.
* Click back on the <H2> tag to select it, and the CSS Property task pane updates to reflect this new selection. You can choose to scroll down the list of properties to see those affected, or alternatively, you can press the **Summary** button to have them all shown at once.  
  
* In this view we can clearly see the implication of the rules we have created. We can see in this instance that this <h2> tag has a font of “Arial” applied to it that is overriding another family “verdana” as indicated by the red strike thru.
* Hovering over each rule in the Applied Rules section shows a tool tip with the font’s rules.  
  
* Clicking to highlight each rule will show you the properties it affects along with any that are being overridden.

In this section, we have only grazed the surface of the power of the CSS features of Expression Web. We encourage you to experiment further and to explore all the online help for the various features we have covered. We will now move on to data presentation and the creation of web applications.

### 14. Rich Data Presentation

* Expression Web allows the designer to work with XML derived data on the design surface in conjunction with the XLST technology. To follow this section requires access to an XML file. We will use Expression Web’s import feature to download such a file from the web.
* Choose **File > Import > File…** then press the **From Site** button.
* This wizard can be used in many ways to import content from a variety of places, not just a single file like this.
* Choose the **HTTP** option and enter the following location or copy and paste from the PDF if you are able:

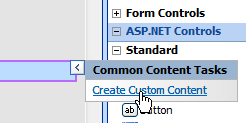
<http://fabrikam.members.winisp.net/products/products.xml>   


* Press the **Next >** button three times (accepting all the defaults) and then the **Finish** button. You should then have the xml file in your folder list ready to use. Let’s create a new file to host our xml data.
* **File > Save** and **File > Close** all open files until you can see the folder list in the center of the screen as well as the task pane.
* To create a new document based on our master file, choose **File > New… > Create From Master Page…** and then browse for “site.master”, click **Open** and then **OK**.
* The page looks the same as the master, but of course all the elements we have worked on thus far are locked.
* Choose **File > Save** and name the file as “data.aspx” - .aspx being the file extension for an ASP.NET page.
* We can see our content area on the page, but we still cannot access it. This is because by default even user content areas are locked. To the right of this area is a small arrow point to the right. This is the Common Tasks Menu that appears on all controls on the design surface. Click on it to open it and choose **Create Custom Content**.  
    
  This unlocks that region allowing us to enter any content we wish.
* Now we are able to click inside the box to activate it. You can check this by making sure your cursor is flashing inside the box and the <asp:content#Content1> tag is highlighted in orange in the tag breadcrumb trail at the top.
* From the folder list task pane, drag and drop the products.xml file into the content area.  
  
* Immediately the content of the XML file is rendered as a table on the design surface and another common task menu has appeared at the upper right hand side, offering us options to fine tune the look and feel of the data.
* Choose the **Edit Columns** button and remove the image column and click OK.  
  
* Now choose the **Sort** button and add the Name column to sort by and click OK.  
  
* Finally choose the **Change Layout…** button and view all the automatic styles that can be applied to represent the data. Click **Cancel**.
* To reveal what Expression Web has been doing during these data operations, choose **File > Save**. The program immediately prompts you to save what has up till now been a hidden file, an XSL transform document. Click **OK** to save the document in the default location.
* Double click the document “productsA.xsl” in the Folder task pane and you can see what has been created for you automatically. It is this document that translates the original XML into browser readably HTML. All subsequent design changes you make on the design surface, such as styling the text, applying CSS rules etc. get automatically written back to this file. If you wish to use the file elsewhere in the site, the XSLT document can be dragged and dropped onto any other XML control.

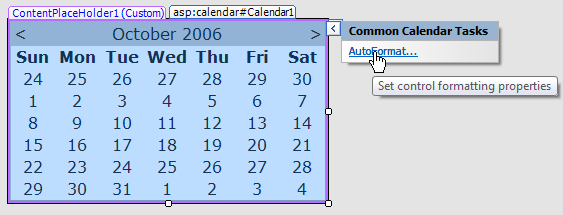
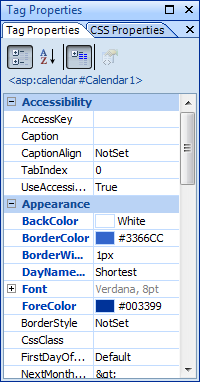
### 15. ASP.NET 2.0

ASP.NET is an entire subject in itself and Microsoft has a wealth of resources online to discuss the technology and the value proposition of using ASP.NET as a platform for developing web applications. For access to this material please visit [www.asp.net](http://www.asp.net)

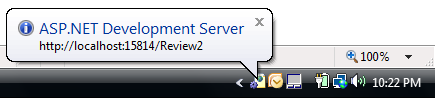
This topic will only deal with the designer’s ability to work with ASP.NET controls.

* Expression Web allows designers access to the range of controls available with ASP.NET 2.0 – in fact Expression Web is the first visual design tool to provide support for this updated version.
* Create a new file by choose **File > New… > Create From Master Page…** and then browse for “site.master”, click **Open** and then **OK**. This gives us another page based on our site’s master file.
* Choose **File > Save** and save the file as “controls.aspx”.
* Use the Common Content Tasks popout to unlock the user region on the page by clicking on it to open it and choosing **Create Custom Content**.  
  
* Using the Toolbox task pane, drag any controls you wish onto the design surface. The calendar, for example, places a fully functioning calendar onto the page. Drag a calendar control onto the user region.

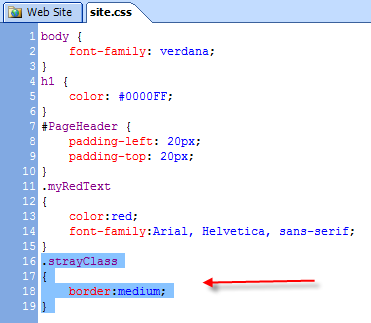
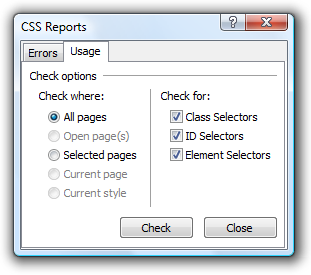
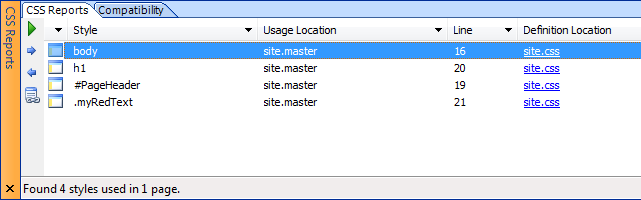
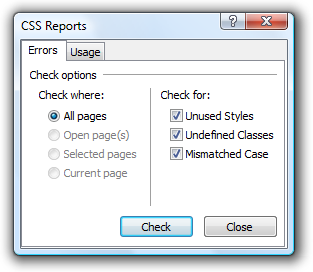
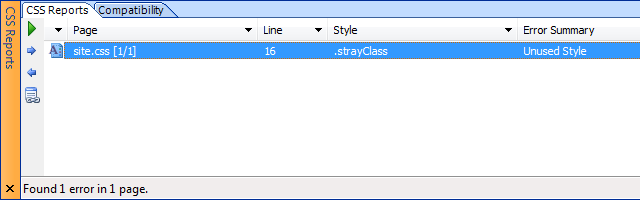
### 16. Control Common Task Popout

* As we’ve already seen a few times, each control has an associated Common Task Popout attached to it, the function of which is unique to each control. What they all have in common however, is the means to allow the designer to set many of the control’s properties from straightforward dialog boxes without having to know the specific syntax required. This syntax would however be available in code view through IntelliSense.
* Open the Common Calendar Tasks popout and chose in this case the only option, **AutoFormat**.  
  
* From the list, choose **Colorful 2**. As noted, this is simply a preset collection of properties for this control. You can view these settings by inspecting the Tag Properties task pane. Each control exposes a different and unique set of properties appropriate to the functionality of that control. This will often include the ability to style controls with CSS classes.  
  
* Please feel free to explore the range of controls in conjunction with the information provided on [www.asp.net](http://www.asp.net).
* As you do so, keep in mind that in Expression Web we allow the designer to use and style the controls, thereby integrating them into the overall look and feel of the site. The workflow would usually be that the project would pass across to developers who would open it in Visual Studio 2005 exactly as saved by Expression Web and continue working with the ASP.NET elements, adding additional server functionality as required. They can achieve this without damaging the design of the site. Indeed, it will often be the case that the site will undergo many iterations between design and development – between Expression Web and Visual Studio 2005.

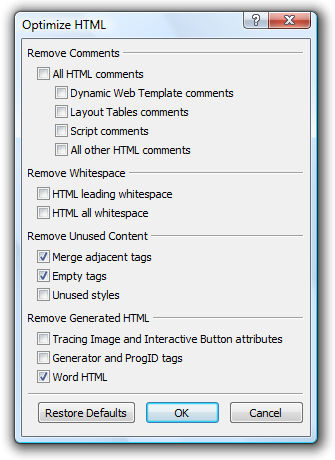
### 17. ASP.NET Development Server

* To be able to view ASP.NET pages within the browser, requires them to be served by an Internet Information Services (IIS) webserver. The designer may well have that installed on their PC, especially if they are using Windows Vista.
* If for any reason they do not have access to a local webserver, Expression Web installs the ASP.NET Development Server, a small, lightweight server that requires no configuration and requires little memory to function. You can see this in action, by viewing the page you have created in your browser using **File > Preview In Browser > Your main browser**.  
  

### 18. Reporting

* Ensuring that your site functions correctly can be achieved by passing the completed site, or indeed the site at any stage, through Expression Web’s extensive range of reports. There are many and varied options you can choose with the reports, in the areas of Accessibility, Compatibility and CSS. Before running the CSS reporting, let’s actually add an error to our site.
* From the Folder List open “site.css” and to the end of the file, add the following unused class:  
    
  .strayClass  
  {  
   border:medium;  
  }  
    
  
* Choose **File > Save** and then **File > Close**.
* Let’s run the CSS usage report by choosing **Tools > CSS Reports…** and then setting the following options:  
  
* This returns a report in the newly opened Results task pane detailing all the styles we have used and where they have been used.  
  
* Typically during the creation of a site, we would expect to end up with many unused styles that can be cleaned up before site deployment. These can be identified using the CSS Error report by choosing **Tools > CSS Reports…** and then setting the following options:  
  
* This report correctly spots our unused class.  
  
* Lastly, let’s run the site through the Accessibility report, by choosing **Tools > Accessibility Reports…**
* This report is highly configurable to check varying levels of conformity with either Section 508, WCAG1 and WCAG2 – or any combination.
* If you run the report with all options selected, you get a wide range of results all of which have specific hyperlinks through to reference material online to research the cause of the error.

### 19. HTML Optimization

* More often than not, final deployment of the site would be from Visual Studio after the design has undergone further application development. If required however, sites can be deployed using Expression Web’s upload and synchronization features. In conjunction with this feature, files can be passed through an HTML optimizer. This can be set to remove all unnecessary information from the file that isn’t required to render the page in the web browser. This can include commenting, additional white space. Choose **Tools > HTML Optimization…** (if the option is not available, open any of the site’s pages to activate it.)  
  

### 20. Extensibility

* Finally, Expression Web has a rich extensibility API that allows 3rd parties to create add-ins that extend the functionality of the program. These would be added using **Tools > Add-Ins…**
* These add-ins will be available through the Microsoft Expression website at [www.microsoft.com/expression](http://www.microsoft.com/expression)

We hope that these short exercises have helped to introduce you to the many and varied design options within Expression Web. We have only touched on some of the features and we hope you will continue to explore the program and help content.

# Product and Company Facts

## PR Contacts

Melissa Hovis [[mhovis@wagged.com](mailto:mhovis@wagged.com?subject=Expression%20Graphic%20Designer%20Reviewer%20Guide%20Follow%20Up)]  
Waggener Edstrom Strategic Communication  
+1 503 443 7127

## System Requirements

The following hardware and software are required for Expression Web:

* PC with Pentium 700-MHz or faster processor
* Windows® XP Service Pack 2 or later or Microsoft Windows Server® 2003 Service Pack 1 or later operating system
* 512 MB of RAM or more
* 1.5 GB of available hard disk space; a portion of this disk space will be freed after installation if the original download package is removed from the hard drive
* CD-ROM or DVD-ROM drive
* 1024×768 or higher resolution monitor
* Internet functionality requires Internet access (fees may apply); Network access required
* Broadband connection, 128 kilobits per second (Kbps) or greater, for download and activation of products
* Actual requirements and product functionality may vary based on your system configuration and operating system

## Company Headquarters

Microsoft Corporation  
One Microsoft Way  
Redmond  
WA 98052-6399  
USA

Microsoft, Microsoft Expression, Windows and Windows Vista are either registered trademarks or trademarks of Microsoft Corp. in the USA and/or other countries.

Other product and company names herein may be trademarks of their respective owners.

© 2006 Microsoft Corp. All rights reserved.

The information contained in this document represents the current view of Microsoft Corp. on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.