

# Chapter 1: Exploring Visual Studio Extensions

---

## *In this Chapter:*

- ✓ Getting the most out of Visual Studio now
- ✓ Building the next generation of Web sites with Atlas
- ✓ Looking ahead to the future of Visual Studio

**T**he name of this chapter should be “Getting Everything You Can Out of Visual Studio 2005 Now and in the Future.” Since Visual Studio’s release in Fall 2005, Microsoft has released a ton of free add-ons you can download from the Web.

## *Extending Visual Studio*

There are many opportunities to extend Visual Studio. Some are add-ons from Microsoft; others are cool toys created by the community. In this section, I list a few of both — and share some additional resources with you, including how to get hold of trial versions of development tools.

### *Tools for administration and troubleshooting*

One of the best things about add-ons is that all you have to do to get them is download them. Some help you administer and troubleshoot development projects; others provide new tools for programming Vista and Office 2007.

Administrative and troubleshooting downloads available from Microsoft include the following:

- ◆ **MSBuild Toolkit** — Use the MSBuild Toolkit to develop solutions that target the .NET 1.1 Framework using Visual Studio 2005. You can download the toolkit at

[www.codeplex.com/Wiki/View.aspx?ProjectName=MSBee](http://www.codeplex.com/Wiki/View.aspx?ProjectName=MSBee)

- ◆ **Team Foundation Server Admin Tool** — As an administrator, add users to Team Foundation Server through a single user interface. Download the tool at

[www.codeplex.com/Wiki/View.aspx?ProjectName=TFSAAdmin](http://www.codeplex.com/Wiki/View.aspx?ProjectName=TFSAAdmin)

- ◆ **Managed Stack Explorer** — Investigate application hangs, using the Managed Stack Explorer. Download it at

[www.codeplex.com/Wiki/View.aspx?ProjectName=MSE](http://www.codeplex.com/Wiki/View.aspx?ProjectName=MSE)

- ◆ **Web Application Projects** — Create ASP.NET 2.0 Web sites using the familiar project model of ASP.NET 1.1. Download at

<http://msdn.microsoft.com/asp.net/reference/infrastructure/wap/default.aspx>

- ◆ **Web Deployment Projects** — Create a project to deploy your ASP.NET 2.0 Web sites. Download at

<http://msdn.microsoft.com/asp.net/reference/infrastructure/wdp/default.aspx>

- ◆ **CSS Control Adapter Toolkit** — Take control of the elements rendered by Web server controls, using Cascading Style Sheets. The toolkit makes it possible to use CSS to override many server controls' default behavior of using <table> tags to render themselves to the browser. Download the toolkit at

[www.asp.net/cssadapters/](http://www.asp.net/cssadapters/)



If you're not that familiar with using CSS for Web-page layout, I suggest you pick up a copy of *CSS Web Design For Dummies*, by Richard Mansfield (Wiley Publishing, Inc.). The book goes way beyond your basic CSS styling and shows you how to create multi-column Web pages without using HTML tables.

- ◆ **ASP.NET Providers source code** — Download the source code for all the built-in providers in ASP.NET 2.0, such as Membership and Site Navigation, at

<http://download.microsoft.com/download/a/b/3/ab3c284b-dc9a-473d-b7e3-33bacfcc8e98/ProviderToolkitSamples.msi>

- ◆ **Developer Highway Code** — Access and use an e-book that features guidance and security checklists for .NET 1.1 and 2.0. It's a 147-page PDF file about writing secure code, brought to you by the Microsoft Patterns & Practices group. Download the e-book at

[www.microsoft.com/uk/msdn/security/dev\\_highway.msp](http://www.microsoft.com/uk/msdn/security/dev_highway.msp)

- ◆ **Spec# Programming System** — Use the Spec# programming language to extend design-by-contract features to C#. Get more information at

<http://research.microsoft.com/specsharp/>



The MSBuild Toolkit, Team Foundation Server Admin Tool, and Managed Stack Explorer are part of the Visual Studio Power Toys. They are released under Microsoft's Shared Source initiative. You can read more about Shared Source and find additional downloads at

[www.microsoft.com/resources/sharedsource](http://www.microsoft.com/resources/sharedsource)



You may have noticed that several of the resources in the list are on the CodePlex Web site. CodePlex is a Web site that Microsoft dubs as a “new collaborative development portal.” CodePlex was formally launched in June 2006 with 30 open-source projects for developers using Microsoft platforms. Be sure to check out the CodePlex Web site at [www.codeplex.com/](http://www.codeplex.com/).

## *New programming tools for Vista and Office*

With the release of Windows Vista and Office 2007 coming right on the heels of the Visual Studio 2005 release, Microsoft is making available a number of tools that target these new programming models. At the time of this writing, many of these downloads are betas or community technology previews. I suspect that final releases will be available from the Microsoft Web site. Here are a few resources you should check out:

- ◆ **Extensions for Windows Workflow Foundation** — Provides support for building applications that use Windows Workflow Foundation.
- ◆ **Orcas WinFx Development Tools** — Supports building applications that target WinFx with XAML support and project templates.
- ◆ **.NET Framework version 3.0** — Provides the libraries you need to target all the new features of Windows Vista.
- ◆ **Windows Software Development Kit** — Includes documentation and samples that demonstrate the new features of Windows Vista.

You can find the latest version of these resources on the Vista Web site at <http://msdn.microsoft.com/windowsvista/downloads/products/getthebeta/>.

See Chapter 3 in this mini-book for more details on Windows Vista and WinFx.

Many resources for programming Office 2007 are available at the Office Developer Center on MSDN. Some of these resources work with Visual Studio Professional 2005 while others require Visual Studio Tools for Office. Visit the Tools and Technologies page at <http://msdn.microsoft.com/office/future/tools/>.

## *Development resources and new server products*

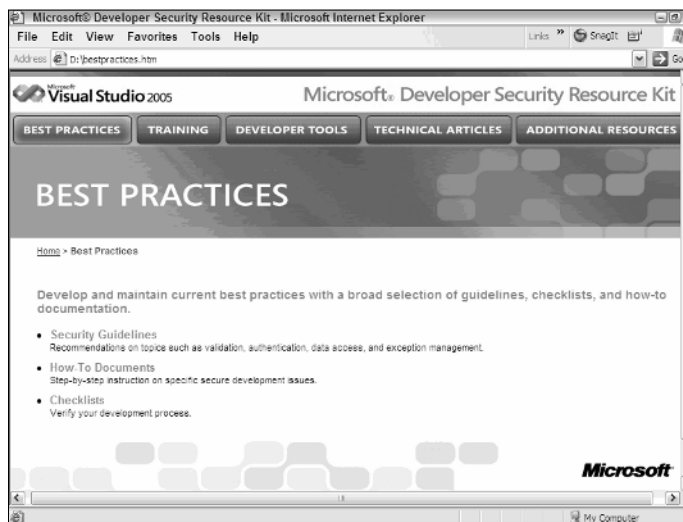
Microsoft has created a Web site where you can order several resources on DVD. The resources are free; you pay shipping and handling. The following resources are available for order at [www.tryvs2005.com](http://www.tryvs2005.com):

- ◆ **Visual Studio 2005 Team Suite 180-day trial** — Includes 180-day trial of SQL Server 2005 Enterprise Edition and Team Foundation Server.
- ◆ **Visual Studio 2005 Professional Edition 90-day trial**

- ◆ **SQL Server 2005 Enterprise Edition 180-day trial**
- ◆ **Microsoft Developer Security Resource Kit** — Includes articles, whitepapers, webcasts, and code samples. Also includes an e-book sample from *Writing Secure Code* by Michael Howard.
- ◆ **Visual Studio 2005 Accelerator Kit** — Includes guidance for migrating to .NET 2.0, code samples, articles, and a free e-book called *Refactor!*.
- ◆ **Microsoft Patterns & Practices (January 2006 release)** — Includes several patterns & practices guides and the complete January 2006 release of the Enterprise Library.
- ◆ **Windows Mobile 5.0 Developer Resource Kit** — Includes resources to get started developing applications that target Microsoft's next mobile platform.

The kits are guided tours of the resources available for a given topic, conveniently available on a DVD. Figure 1-1 shows an example of the Security Resource Kit.

**Figure 1-1:**  
Order  
kits from  
Microsoft  
that  
conveniently  
bundle  
resources  
on a given  
topic.



In addition to its kits and other development resources, Microsoft is releasing a slew of new enterprise-server products. Many of these products have been developed closely with the architecture of the .NET Framework, and are integrated with Visual Studio 2005:

- ◆ **Commerce Server 2007** — A server product for creating out-of-the-box e-commerce Web sites that integrate with a company's back-end systems. Commerce Server 2007 releases in the Fall 2006 and includes a starter site. Commerce Server was built using ASP.NET 2.0, and uses Visual Studio 2005 to build and customize Commerce Server sites.
- ◆ **SharePoint** — There are two SharePoint products:
  - Windows SharePoint Services version 3 provides the collaboration services of SharePoint sites.
  - Microsoft Office SharePoint Server 2007 is a portal product that's part of Office 2007.

Microsoft intends to release a set of SharePoint extensions for Visual Studio about six months after the release of Office 2007. These extensions add a SharePoint project template to Visual Studio.

In the meantime, you can download Software Development Kits for Windows SharePoint Services version 3 at [www.microsoft.com/downloads/details.aspx?familyid=05E0DD12-8394-402B-8936-A07FE8AFAFFD&displaylang=en](http://www.microsoft.com/downloads/details.aspx?familyid=05E0DD12-8394-402B-8936-A07FE8AFAFFD&displaylang=en). The SharePoint Server 2007 SDK can be found at [www.microsoft.com/downloads/details.aspx?familyid=6D94E307-67D9-41AC-B2D6-0074D6286FA9&displaylang=en](http://www.microsoft.com/downloads/details.aspx?familyid=6D94E307-67D9-41AC-B2D6-0074D6286FA9&displaylang=en).

- ◆ **Internet Information Services 7** — The next version of IIS has integrated support for ASP.NET. IIS7 has its own dedicated Web site at [www.iis.net](http://www.iis.net) where you'll find technical resources, articles, blogs, and other community resources. Write your own modules — using the .NET Framework to target the IIS application programming interface — or become more knowledgeable about how to configure and tweak IIS. The Downloads section features an IIS7 Managed Module Starter Kit for C# or C++.

Other fun items you might want to download from the community include

- ◆ **Spell checker for ASP.NET** — Spell-checks the text you add to your ASP.NET Web pages. This was written by one of the developers on Microsoft's Web Development Tools Team in his spare time. In other words, the tool isn't supported by Microsoft. You can download the spell checker at <http://blogs.msdn.com/mikhailarkhipov/default.aspx>.
- ◆ **NDepend** — Code analyzer that visually displays your source code's architecture and allows you to apply many code metrics against your source code. Download for free at [www.ndepend.com](http://www.ndepend.com).
- ◆ **MSBuild Community Tasks Project** — An open-source project that's building a library of common MSBuild tasks. You'll find just about everything you need to automate your nightly builds. Visit the project's Web site at <http://msbuildtasks.tigris.org/>.

- ◆ **RSS Toolkit** — Created by a member of the ASP.NET development team for consuming and publishing RSS feeds. Download the toolkit at <http://blogs.msdn.com/dmitryr/archive/2006/03/26/561200.aspx>.

Microsoft has embraced blogging. A *blog* (short for *Web log*) is a Web site where a blogger posts entries and visitors post comments on the blog entries. Here's a short list of blogs you might want to keep tabs on:

- ◆ **Web Development Tools Team Blog** — These are the folks that develop Visual Web Developer. Read their blog at <http://blogs.msdn.com/webdevtools/>.
- ◆ **S. “Soma” Somasegar’s blog** — Soma is the corporate vice president for the Microsoft Developer Division. His blog always features the latest news about Visual Studio 2005. Find his blog on MSDN at <http://blogs.msdn.com/somasegar/>.
- ◆ **Developer Division Customer Product Lifecycle Experience Team (DDCPX) Team Blog** — These are the folks responsible for bringing aftermarket solutions — such as the Visual Studio Power Toys and software-development kits (SDKs) — to you. Find them at <http://blogs.msdn.com/ddcpxblog/>.
- ◆ **MSBuild Team Blog** — MSBuild is the build engine for Visual Studio. Read the team blog at <http://blogs.msdn.com/msbuild>.

You can find more Microsoft blogs at <http://blogs.msdn.com/> and [www.microsoft.com/communities/blogs/](http://www.microsoft.com/communities/blogs/).

MSDN has created a number of developer centers that provide links to blogs, downloads, articles, and other resources of interest to developers. Some centers you might want to visit include:

- ◆ **Security Developer Center** — <http://msdn.microsoft.com/security/>
- ◆ **Visual Studio Developer Center** — <http://msdn.microsoft.com/vstudio>
- ◆ **Data Access and Storage Developer Center** — <http://msdn.microsoft.com/data/>
- ◆ **Smart Client Developer Center** — <http://msdn.microsoft.com/smartclient/>
- ◆ **Office Developer Center** — <http://msdn.microsoft.com/office/>

Find even more developer centers on MSDN at <http://msdn.microsoft.com/developercenters/>.

## Exploring AJAX and the Atlas Library

Atlas is the codename for a framework created by Microsoft to provide the AJAX approach to developing user interfaces for Web applications. AJAX stands for Asynchronous JavaScript and XML. Developers use a combination of client-side JavaScript and server-side programming to create a Windows-like user experience. AJAX offers an important benefit: the browser and the server can communicate without a postback; from an end-user standpoint, that means no screen flash while the page refreshes. Popular Web sites using AJAX include

- ◆ Google Suggest
- ◆ Google Maps
- ◆ Flickr
- ◆ Gmail
- ◆ Outlook Web Access
- ◆ Yahoo! Mail Beta



To get a feel for some of the things you can do with AJAX, go to the demos page at <http://openrico.org>.

To get a feel for the underlying technologies of AJAX and what it takes to AJAX-ify your Web sites using technologies besides .NET, pick up a copy of *Ajax For Dummies*, by Steve Holzner (Wiley Publishing, Inc.).



The technologies used in AJAX are nothing new. They've been around for a long time; only recently have folks applied the name AJAX to this style of programming. The technology that underlies AJAX is the object `XMLHttpRequest`. This one object makes it possible to send data back and forth between the client and server without a page refresh.

Atlas is Microsoft's version of AJAX. It consists of:

- ◆ **Client-side library** — Atlas provides object-oriented access to the client-side features you would usually have to write in JavaScript. With Atlas, you can apply what you know about object-oriented programming in .NET without diving into JavaScript.
- ◆ **Server controls** — Server-based controls similar to existing Web server controls, such as buttons and text boxes.

Atlas makes extensive use of Web services to provide the server-side features of the client-server interaction of AJAX. It's quite common for developers to

wire up the elements of their user interfaces to Web services that provide data (such as specific items for a drop-down list). Of course, you aren't limited to consuming only Web services you create. You can also use an Atlas-enabled Web page to connect to third-party Web services.



Web sites that use content from other sources are called *mashups*. Mashups often utilize several technologies including Web services, RSS, and AJAX. Mashups are a driving force in the *Web 2.0* movement (the term often used to refer to the next generation of Web applications).

## Installing and setting up Atlas

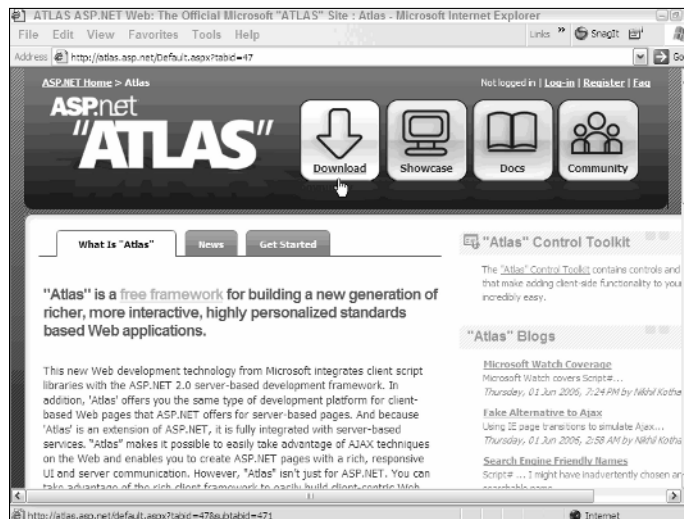
Atlas is essentially a library provided by Microsoft. To use Atlas, you must first download and install the library. The download includes Visual Studio project templates for Visual Basic and C#. To download Atlas, follow these steps:

1. Browse to the Atlas Web site at <http://atlas.asp.net/>.

The Atlas Web site features a Get Started tab that walks you through downloading, installing, and using Atlas.

2. Click the Download button in the Web site's header, as shown in Figure 1-2. The Atlas Downloads page appears.

**Figure 1-2:** Click the Download button on the Atlas Web site to get the latest release.





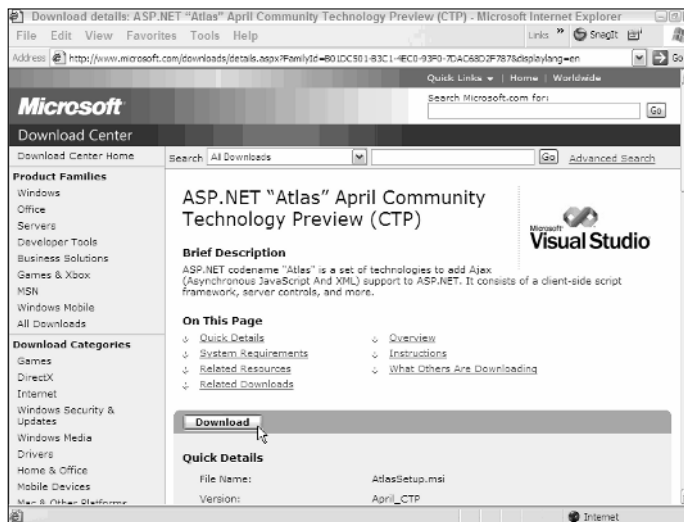
**3. Click the link for the latest release of Atlas. You are taken to the Download Center on Microsoft's Web site.**

The examples in this chapter use the April Community Technology Preview (April CTP).

**4. Click the Download button to download the file, as shown in Figure 1-3.**

Your browser may prompt you to download or run the file depending on how your browser is configured. You can choose to run the file now or save it to your hard drive and execute it later.

**Figure 1-3:**  
Click the  
Download  
button at the  
Microsoft  
Download  
Center to  
download  
the setup  
file.



The instructions and screenshots in this section use the April CTP version of Atlas. Later releases of Atlas may look slightly different. Generally speaking, you should expect to successfully install Atlas by simply following the installation wizard. In the event you have problems, see the official Atlas Web site at <http://atlas.asp.net/>.

To install the file you downloaded, perform these steps:

**1. Execute the setup file from where you saved it, or directly from the browser.**

The Setup Wizard starts.

**2. Click the Next button.**

The license agreement appears.

**3. Read the license agreement; click *I agree* if you agree with the terms of the license.**

**4. Click the Next button.**

The Install Project Template step appears.

**5. Accept the option to install Visual Studio project templates.**

**6. Click the Next button.**

The Register .aspx File Extension step appears.

**7. Leave the option to register the .aspx file extension checked if you intend to use Atlas to communicate with external Web services.**

**8. Click the Next button.**

The Confirm Installation step appears.

**9. Click the Next button.**

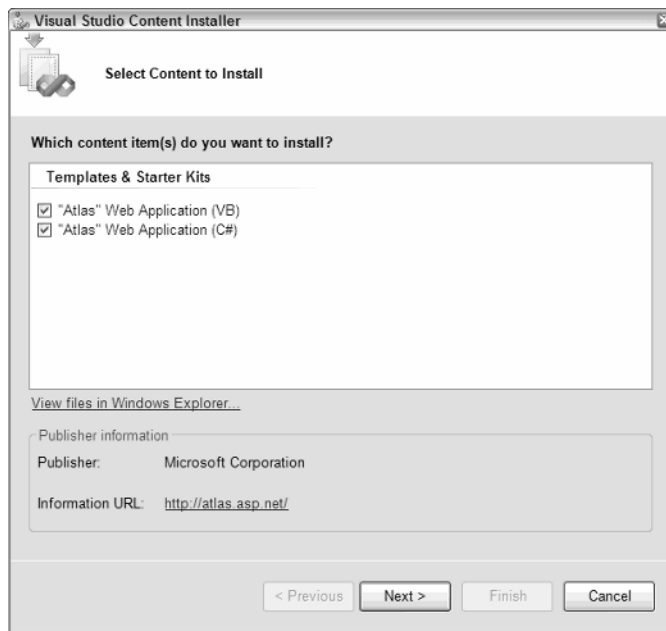
The installation begins. After Atlas installs, the Visual Studio Content Installer appears.

**10. Accept the default entries in the content installer to install Atlas project templates in Visual Studio, as shown in Figure 1-4.**

---

**Figure 1-4:**  
The Visual  
Studio  
Content  
Installer  
prompts  
you to  
install Atlas  
project  
templates.

---



**11. Click the Next button.**

A confirmation step appears.

**12. Click the Finish button.****13. Click the Close button after the wizard confirms that installation is complete.**

You create a new Atlas project using the New Web Site window in Visual Studio. The Atlas project templates provided for Visual Studio wires up Atlas to an ASP.NET Web site for you by doing the following:

- ◆ Adding a reference to the `Microsoft.Atlas.Web.Atlas.dll` assembly.
- ◆ Adds a `ScriptManager` component.
- ◆ Adds Atlas configuration settings to the site's `web.config` file.

Before you dive in and fire up your first Atlas Web site, I suggest you download the Atlas Control Toolkit — which includes controls built using Atlas. The toolkit allows you to start using the features of Atlas without digging into the details of how the architecture works. To download and install the toolkit, follow these steps:

**1. Click the Atlas Control Toolkit link on the Atlas Downloads page of the <http://atlas.asp.net> site.**

The file `AtlasControlToolkit.exe` downloads to your computer.

Alternatively, you can download the toolkit at

```
www.codeplex.com/Wiki/View.aspx?ProjectName=
AtlasControlToolkit
```

**2. Execute the `AtlasControlToolkit.exe` file.**

You are prompted for a location to unzip the toolkit's contents.

**3. Enter the name of the file directory in which you want to unzip the file.****4. Click the Unzip button.**

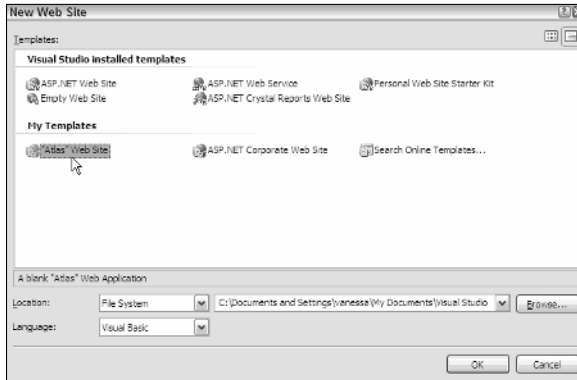
The toolkit unzips and launches a Web-based setup walkthrough that includes instructions for installing Atlas, configuring Visual Studio 2005 to use the toolkit, and samples of the controls in the toolkit.

## ***Adding the Atlas Toolkit controls***

Before you can start using the controls, you must add them to the toolbox in Visual Studio. To add the Atlas Toolkit controls to the Visual Studio 2005 toolbox:

**1. Create a new Web site using the Atlas Web Site template, as shown in Figure 1-5.**

The Atlas Web Site template is installed when you install Atlas. See Chapter 4 in Book III for more information on creating Web sites.



**Figure 1-5:**  
The New  
Web Site  
window.

**2. Press Ctrl+Alt+X to open the Toolbox.**

**3. Right-click on the Toolbox and click Add Tab.**

A new tab appears in the toolbox.

**4. Type Atlas Control Toolkit as the name for the new tab.**

**5. Right-click in the tab you just added.**

A shortcut menu appears.

**6. Click Choose Items from the shortcut menu.**

The Choose Toolbox Items window appears.

**7. Click the Browse button.**

The Open window appears.

**8. Browse to the location where you unzipped the Atlas Control Toolkit in Step 4 of the preceding set of steps.**

**9. Open the SampleWebSite folder.**

**10. Open the Bin folder.**

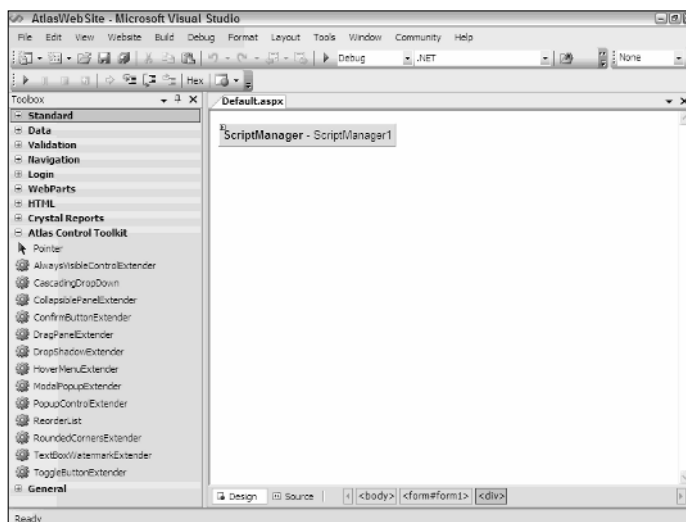
**11. Click the file AtlasControlToolkit.dll and click the Open button.**

A link to the dll file appears in the Choose Toolbox Items window.

**12. Click OK.**

The controls appear in the Atlas Control Toolkit tab of the toolbox. As with all controls in the toolbox, these controls are visible only if a Web page is open in the designer, as Figure 1-6 shows.

**Figure 1-6:**  
The Atlas controls appear in the Atlas Control Toolkit tab in the toolbox.



The controls provided in the Atlas Control Toolkit aren't controls in the strict sense that a text box or a button are controls. Rather, they are control *extenders* — they add properties to existing controls.



The toolkit provides templates for creating your own extenders. Refer to the toolkit's documentation for more information.

To use one of the sample extenders provided in the toolkit, follow these steps:

1. **Select a sample you want to use from the toolkit and determine which controls it extends.**

For example, the `ToggleButton` extender works with the `CheckBox` control. (Refer to the toolkit's documentation for information on each of the control extenders.)

2. **Add Web server controls to your Web page that are extended by the Atlas toolkit to your Web page.**

For example, add the buttons, panels, check boxes, or other controls that are extended by the toolkit.

3. **Add the control extender from the Atlas Control Toolkit tab in the toolbox.**

4. **Open the Properties window for the control you're extending.**

5. **Expand the extended properties for the control, and then enter values for the properties.**

For example, the toolkit includes a `CollapsiblePanel` extender that you can use to collapse and expand a panel. Follow these steps to use the `CollapsiblePanel` extender:

1. Drag a `LinkButton` control onto the Web page.
2. Drag a `Label` control inside the `LinkButton` control.
3. Drag a `Panel` control onto the Web page.
4. Drag a `Label` control inside the `Panel` control.

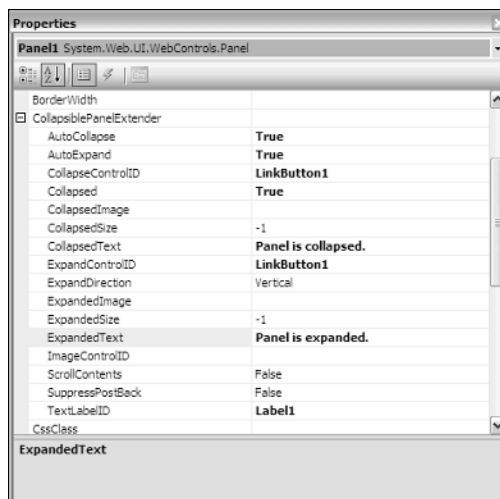


You can place any server control inside a `Panel` control. Most often, the `CollapsiblePanel` extender is used with a panel to collapse and expand a list or a grid view.

5. Set the label's `Text` property to `Hello world`.
6. Drag a `CollapsiblePanelExtender` onto the Web page.
- The extender adds a set of extended properties to the panel control added in Step 3.
7. Click the `Panel` control you add in step 3 and press F4 to view the panel's properties.
8. Expand the `CollapsiblePanelExtender` properties.
9. Set the extended properties as shown in Figure 1-7.

Note the `LinkButton` you create in step 1 is set as the `CollapseControlID` and the `ExpandControlID`. The label created in Step 2 is the value for the `TextLabelID` property.

**Figure 1-7:**  
Set the  
properties  
extended by  
the Atlas  
Control  
Toolkit.



Press Ctrl+F5 to run the Web site. The page displays a link button that toggles between “Panel is collapsed” and “Panel is expanded” as you click the link.

## Looking Ahead to the Next Visual Studio

Microsoft has plans for two future versions beyond Visual Studio 2005:

- ◆ **Codename Orcas** — Orcas is supposed to allow developers to tap into the new features of Windows Vista and Windows Longhorn Server. Orcas is expected to release in the second half of 2007.
- ◆ **Codename Hawaii** — Hawaii is supposed to be a complete rebuild — from the ground up — of Visual Studio.

Keep up to date with the future offerings for Visual Studio at <http://msdn.microsoft.com/vstudio/future/>. You'll find previews for future versions of Visual Studio, as well as extensions for the current version.

### Orcas features

Given that Visual Studio 2005 released late and Office 2007, Windows Vista, and Longhorn Server all slipped their schedules, it seems that Visual Studio 2005 is stealing much of the thunder from Orcas. Many of the tools slated for Orcas are being made available for Visual Studio 2005.

Some of the features you can expect to find in Orcas include:

- ◆ **WinFX** — A library for accessing the features of the Windows Vista operating system. WinFX is a framework similar to the .NET Framework.
- ◆ **LINQ** — Extensions to the .NET Framework that support the next generation of data access in .NET applications.
- ◆ **Visual Basic 9.0** — Read more about the features of the next version of Visual Basic at <http://msdn.microsoft.com/vbasic/future/>.
- ◆ **C# 3.0** — Read more about the proposed changes to the C# language at <http://msdn.microsoft.com/vcsharp/future/>.
- ◆ **Visual Tools for Office** — Updated development tools for taking advantage of Office 2007.
- ◆ **XAML** — XAML, pronounced *zamel*, is a new declarative language for describing user interfaces. XAML makes it possible for designers to use graphic design tools that describe the user interface in an XML file. Developers can easily open the file in Visual Studio and work with the elements of the user interface.



A key player in XAML is Microsoft's new line of graphic-design tools called Expression. Read more about Expression at [www.microsoft.com/products/expression/](http://www.microsoft.com/products/expression/).

## ***The LINQ Project***

LINQ is short for Language Integrated Query. In essence, LINQ creates a query language right inside the .NET Framework and the programming languages of the .NET Framework — mainly Visual Basic and C#. LINQ can be used to query any kind of data, but its primary targets are XML and SQL data.

The LINQ query architecture defines a generic set of query operators, but can be extended to handle more specific operators. Two such extensions already exist:

- ◆ **Xlinq** — These XML query operators provide XPath/XQuery features.
- ◆ **Dlinq** — These SQL query operators build on the existing SQL support in the .NET Framework.

The significance of LINQ is that you can write queries using C# or Visual Basic. You can use the standard query operators to query data in arrays or collections and use Xlinq and Dlinq to query XML and SQL data sources, respectively. Instead of specifying queries using strings as is done now, the queries are part of the language syntax.

The LINQ operators use a set of SQL-like syntax. For example, the LINQ specification defines the following operators:

- ◆ Where
- ◆ Select
- ◆ SelectMany
- ◆ Join
- ◆ GroupJoin
- ◆ OrderBy

There are many more operators, but you can see the resemblance to SQL.

## ***Take a test drive with the CTP***

A *community technology preview* (CTP) of LINQ was released in May 2006 for Visual Studio 2005. You download the CTP and read more about LINQ at the Project LINQ Web site at <http://msdn.microsoft.com/data/ref/linq/>.



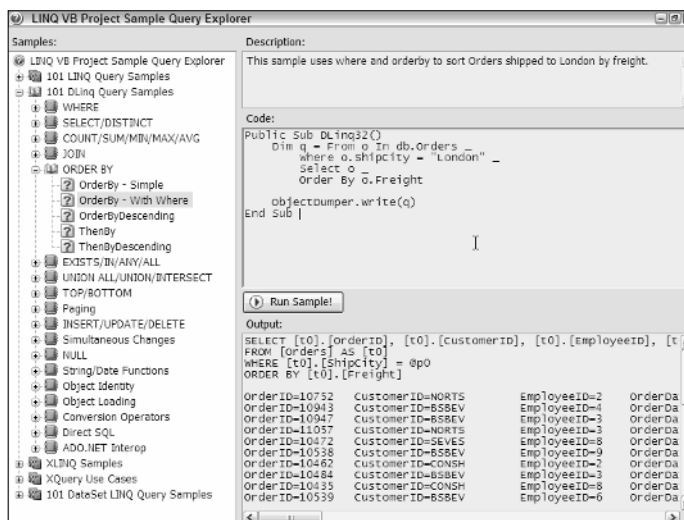


You can find many other CTP downloads for Orcas at [http://msdn.microsoft.com/vstudio/future/ctp\\_downloads/default.aspx](http://msdn.microsoft.com/vstudio/future/ctp_downloads/default.aspx).

The CTP includes code samples and documentation about LINQ for Visual Basic and C#. The CTP installs Visual Studio project templates you can use to create your own LINQ-enabled projects.

The code samples demonstrate all the query operators for LINQ, Dlinq, and Xlinq. The samples also show how to use Dlinq with DataSets. Figure 1-8 shows code sample that demonstrate Dlinq queries.

**Figure 1-8:**  
The code samples demonstrate LINQ's query operators.



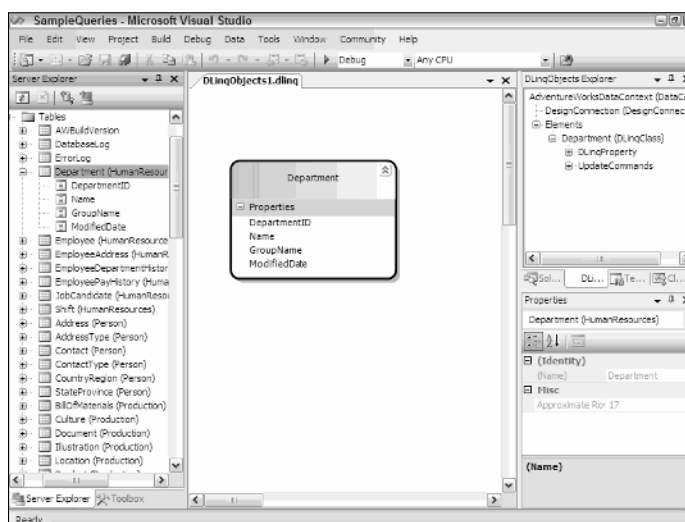
The LINQ CTP includes a Dlinq Designer which is a visual designer for creating Dlinq queries. Essentially, you drag and drop tables from Server Explorer to the designer — and the designer translates relational database objects (such as tables and relationships) into graphic objects. You write Dlinq queries that target the objects. Figure 1-9 shows an example of the Dlinq Designer. The shapes on the designer represent entity classes generated by the designer to represent relational tables.

## A peek at Hawaii

For obvious reasons, not much information is available about the version of Visual Studio after Orcas, codenamed “Hawaii.” It has been reported that Microsoft plans to completely re-architect the product. I interpret that to mean a completely new approach to software development.

My guess is that this next-generation approach to development will allow developers to immerse themselves in modeling their business problem — and in generating code from that model — rather than getting stuck in a “code zone.” Three development resources that I suggest accessing for a closer look are as follows:

**Figure 1-9:**  
The Dlinq  
Designer  
generates  
entity  
classes  
from  
relational  
tables.



- ◆ **Microsoft’s Center for Software Excellence** ([www.microsoft.com/Windows/CSE](http://www.microsoft.com/Windows/CSE)) — A group in Microsoft Research focused on developer productivity at Microsoft.
- ◆ **Intentional Software** ([www.intentionalsoftware.com](http://www.intentionalsoftware.com)) — The company founded by Charles Simonyi to bring to market the next generation of software development. Simonyi is the man behind Microsoft Word and Excel. He also created the Hungarian notation naming convention.
- ◆ **Software Factories** ([www.softwarefactories.com](http://www.softwarefactories.com)) — The idea of being able to mass-produce code from a model is being used now in Visual Studio Team System. Expect to see this concept trickle down to the individual developer in the future.

You can see how Visual Studio Team System is already using models to generate code in the next chapter.