InnerWorkings Catalog

A complete guide to the InnerWorkings .NET training catalog, with prerequisites, learning goals & technical levels for each Drill.
## Contents

InnerWorkings Catalog ......................................................................................... 1

Introduction .............................................................................................................. 1
What the levels mean ................................................................................................. 1
Recommended books ................................................................................................. 1

### Beginning ASP.NET 4.5 in C# ............................................................................. 2
Drill: Designing and Building ASP.NET 4.5 Websites .............................................. 3
Drill: ASP.NET 4.5 Website Structure ......................................................................... 4
Drill: Building ASP.NET 4.5 Pages ............................................................................ 5
Drill: Managing Data in ASP.NET 4.5 ......................................................................... 6
Drill: Securing ASP.NET 4.5 Web Applications ....................................................... 7

### Windows 8 App Development with JavaScript & HTML5 ................................... 8
Drill: Windows 8 App Development Foundation ....................................................... 9
Drill: Windows 8-Style UI Layouts and Surfaces ....................................................... 10
Drill: Windows 8-Style Controls .............................................................................. 11
Drill: Resource Management ................................................................................. 12
Drill: Hardware & Multimedia .............................................................................. 13
Drill: Contracts ........................................................................................................ 14

### Windows 8 App Development with C# and XAML .............................................. 15
Drill: Windows 8 App Development Foundation ....................................................... 16
Drill: Windows 8-Style UI Layouts and Surfaces ....................................................... 17
Drill: Resource Management ................................................................................. 18
Drill: Contracts ........................................................................................................ 19

### Windows Azure .................................................................................................. 20
Drill: Windows Azure Web Sites .............................................................................. 21
Drill: Windows Azure Virtual Machines ..................................................................... 22
Drill: Windows Azure Cloud Services ....................................................................... 23
Drill: Windows Azure Storage Services .................................................................... 24
Drill: Windows Azure Caching ............................................................................... 25
Drill: Windows Azure Service Bus .......................................................................... 26
Drill: Windows Azure & Windows 8 .......................................................................... 27
Drill: Advanced Cloud Services ............................................................................. 28
Drill: Windows Azure Usage Scenarios ..................................................................... 29
Drill: Windows Azure Advanced Storage Service .................................................... 30
Introduction

This document provides a comprehensive overview of the InnerWorkings catalog to help you identify which Drills are of greatest relevance to your specific .NET learning needs.

Each Drill description includes the following:

- Overview – a summary of the Drill and what it will enable you to do
- Audience – a description of the target audience
- What you'll learn – a description of the key learning outcomes
- Core development tasks – a checklist of development related activities
- What you'll cover – a listing of the programming topics covered
- Prerequisites – what you'll need to know before getting started
- Drill details – key information such as system requirements, duration, and level

What the levels mean

InnerWorkings Drills are allocated a rating that indicates the technical level of the topics covered. The overall level classification scheme is as follows:

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>Suitable for beginners and above. This level addresses introductory programming concepts and fundamental principles of development.</td>
</tr>
<tr>
<td>Level 2</td>
<td>Suitable for programmers with basic (or higher) programming skills. This level develops essential skills and experience in the core programming framework.</td>
</tr>
<tr>
<td>Level 3</td>
<td>Suitable for programmers with intermediate to advanced programming skills. This level explores more advanced topics, or probes more deeply into level 2 frameworks.</td>
</tr>
</tbody>
</table>
Beginning ASP.NET 4.5 in C#

In this InnerWorkings Beginning ASP.NET 4.5 program, you'll find a large variety of C# coding challenges that will get you firmly on the road to building and deploying ASP.NET Web applications.

This Developer Program contains the following Drills:

- Designing and Building ASP.NET 4.5 Websites
- ASP.NET 4.5 Website Structure
- Building ASP.NET 4.5 Pages
- Managing Data in ASP.NET 4.5
- Securing ASP.NET 4.5 Web Applications
Drill: Designing and Building ASP.NET 4.5 Websites

ASP.NET is the part of the .NET framework that enables you to build great Web sites and Web applications using HTML, CSS and JavaScript.

In these coding challenges, you'll look at some particular aspects of ASP.NET related to building web pages and using server controls.

What you'll learn
First, you'll learn how to use CSS to style a Web form created using the fieldset and definition list elements.

Then you'll learn how to use some of ASP.NET's built-in server controls, and you'll learn about the role of control state in custom server controls.

In the fourth challenge, you'll see how to improve your coding by separating an application's domain logic from its presentation code.

Prerequisites
To create ASP.NET 4.5 apps, you need Visual Studio 2012 and either Windows 7 or Windows 8. So these will be our minimum requirements too. (Visual Studio Express 2012 for Web edition works fine.)
Drill: ASP.NET 4.5 Website Structure

In this drill you will use some of the features of ASP.NET that allow you to easily create Web applications that have a quality look and feel.

What you'll learn
In these coding challenges you will experience some of the practical aspects of creating consistent web sites and web applications.

With master pages, you can produce a single master page and then define the site's layout controls (header, footer, and navigation bar) a single time. You can then propagate the layout throughout the site. Alternatively, you can create a theme and apply it throughout the web pages of an ASP.NET application.

You will also use the navigation web server controls and built-in SiteMap providers to create a simple web page navigation structure, and you'll programatically modify the site navigation data returned by site-map providers.

Prerequisites
To create ASP.NET 4.5 apps, you need Visual Studio 2012 and either Windows 7 or Windows 8. So these will be our minimum requirements too. (Visual Studio Express 2012 for Web edition works fine.)
Drill: Building ASP.NET 4.5 Pages

In this drill, you will use some of the features of ASP.NET that allow you to create useful and attractive Web pages.

With user controls and ASP.NET AJAX, you can create almost any user interface you need, and the reliability and security of your applications are enhanced greatly with validation. As the sophistication of your Web applications increase, you'll appreciate the features offered by jQuery.

**What you'll learn**
In these coding challenges you will gain experience of building customized and dynamic Web pages.

You will learn how to add ordinary controls and custom properties to your ASP.NET user control, add your control to an .aspx page and set it up to display content. You will perform validation by enforcing mandatory fields and performing data comparison and range checks using ASP.NET.

The ScriptManager and UpdatePanel controls enable AJAX-style partial page updates - you'll learn to use both, and you get to encapsulate controls in an UpdatePanel to perform Ajax requests.

You'll also learn how to use jQuery selectors to find elements on a Web page, and you'll learn how to traverse and update the DOM elements through the use of jQuery functions.

**Prerequisites**
To create ASP.NET 4.5 apps, you need Visual Studio 2012 and either Windows 7 or Windows 8. So these will be our minimum requirements too. (Visual Studio Express 2012 for Web edition works fine.)
Drill: Managing Data in ASP.NET 4.5

In this drill, you will see how to connect your Web applications to a database and build your application’s functionality based on its content.

The concept of building a Web application upon the foundation of a database makes it possible to construct the application with large quantities of data that are not available to you at the time of design and development. A database backend is absolutely standard in almost all modern full-featured websites.

What you’ll learn
In these coding challenges you will gain experience of attaching a Web application to a database, and working with data-bound controls.

You will learn how use a GridView control with an ObjectDataSource control to enable paging, editing and deleting of records, and to execute commands against a SQL CE application via inline SQL.

You’ll learn how to query an entity data model generated from an existing database, how to use the Entity Frame work to perform CRUD operations against the database, and expose an underlying database schema.

Prerequisites
To create ASP.NET 4.5 apps, you need Visual Studio 2012 and either Windows 7 or Windows 8. So these will be our minimum requirements too. (Visual Studio Express 2012 for Web edition works fine.)

### Drill Details

**Status**
Released

**Developer Program**
Designing and Building ASP.NET 4.5 Websites

**Number of Tasks**
6

**Duration**
Research: 3 hours
Coding: 3 hours

**Programming Language**
C#

**System Requirements**
Visual Studio 2012
C#

**Level**
3

**Drill Code**
• D0086D-CS
Drill: Securing ASP.NET 4.5 Web Applications

In this drill you will see how to use a variety of data integrity, user profile and stability features to help you create secure Web applications.

There are many issues related to securing your Web applications from data breaches, whether malicious or unintentional. Some are obvious - managing passwords and user access rights, for example; and others are less obvious - good exception handling reduces the risk of vulnerabilities in an application, which hackers can sometimes exploit.

What you'll learn
In these coding challenges, you'll learn how to use several login server controls and how to configure them to build simple user authentication pages. And you'll learn how to use ASP.NET's role management to securely configure users and roles.

You'll see how to define and store unique user information, using the User Profiles feature of ASP.NET, and you'll implement error handling to throw and catch exceptions of specific types.

Prerequisites
To create ASP.NET 4.5 apps, you need Visual Studio 2012 and either Windows 7 or Windows 8. So these will be our minimum requirements too. (Visual Studio Express 2012 for Web edition works fine.)
Windows 8 App Development with JavaScript & HTML5

This Developer Program contains the following Drills:

- Windows 8 App Development Foundation
- Windows 8-Style UI Layouts and Surfaces
- Windows 8-Style Controls
- Resource Management
- Hardware and Multimedia
- Contracts
Drill: Windows 8 App Development Foundation

**Audience**
This Drill is for .NET developers who want to learn how to create Windows 8-style apps.

**What you'll learn**
You will first learn how to create a Windows 8-style app using JavaScript. You will then learn how to use the Canvas element, how to build app bar surfaces, and how to use the file picker. Other skills you will learn include how to add event handlers to controls, how to modify controls’ content, and how to fill your app with graphics. You will also learn about asynchronous programming using promises and how to monetize an app using the Windows Store.

**Core development tasks**
- Create new projects using the correct templates
- Create live tiles and secondary tiles
- Add content and draw to the app window (canvas)
- Add app bar commands
- Use FilePicker
- Handle control events
- Animate graphics
- Use CSS transforms
- Use promises
- Use trial conversion and in-app purchase in the Windows Store

**Prerequisites**
We assume that you’re already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you’ll also need to be comfortable with JavaScript and HTML5 syntax. Although not essential, experience with Visual C# would also be useful.

---

**Drill Details**

**Status**
Released

**Developer Program**
Windows 8 App Development with JavaScript & HTML5

**Number of Tasks**
6

**Duration**
Research: 3.5 hours
Coding: 3.5 hours

**Programming Language**
JavaScript & HTML5
C#

**System Requirements**
Visual Studio 2012
Windows 8 Runtime

**Level**
2

**Drill Code**
- D0070A-JS
Drill: Windows 8-Style UI Layouts and Surfaces

**Audience**
This Drill is for .NET developers who want to learn how to control the presentation, navigation and behavior of their Windows 8-style apps.

**What you'll learn**
You will first learn how to control the behavior of your app using the app bar and charms. You will then learn how to enable your app to adapt to different view states and screen resolutions. Other skills you will learn include how provide suitable navigation for Windows 8-style apps, and how to reference and display external content. You will also learn how to use live tiles and badges to display app data on the start screen.

**Core development tasks**
- add content to the app window
- add app bar commands
- add settings using charms
- create flexible layouts using CSS
- scale to pixel density and screens
- use snapped and fill views
- use PageControlNavigator to navigate between pages
- reference content in the application and link to external pages
- add live and secondary tiles
- update tiles

**Prerequisites**
We assume that you’re already familiar with the concepts of programming and you've some experience of Visual Studio. For this Drill, you'll also need to be comfortable with JavaScript and HTML5 syntax. Although not essential, experience with Visual C# would also be useful. If you don’t have prior experience of creating Windows 8-style apps, we recommend looking at the Application Development Foundation drill for Windows 8.
Drill: Windows 8-Style Controls

Audience
This Drill is for .NET developers who want to learn about new controls available for Windows 8-style apps, how to bind controls to data, and how to create sophisticated user interfaces on different input devices.

What you'll learn
You will first learn how to use control classes, and how to override styles in the new Windows Library for JavaScript controls. You will then learn about the most common Windows 8-style data bound controls. Other skills you will learn include animation of page- and content-transitions and list-add and -delete operations. You will also learn about how to handle gestures in apps on different Windows 8 devices.

Core development tasks
• Using selection controls: slider, date picker, toggle switch
• Styling controls: control classes, overriding styles
• List view
• FlipView
• Semantic zoom
• App navigation
• Content animations
• Collection animations
• Identify input devices
• Handle gestures

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio. For this Drill, you'll also need to be comfortable with JavaScript and HTML5 syntax. Although not essential, experience with Visual C# would also be useful. If you don’t have prior experience of creating Windows 8-style apps, we recommend looking at the Application Development Foundation drill for Windows 8.
Drill: Resource Management

Audience
This Drill is for .NET developers who want to learn about resource management for Windows 8-style apps, how to manage application resources asynchronously, and application data across files and networks.

What you'll learn
You will first learn about asynchronous programming in JavaScript, focusing on some of the more difficult topics. You will then learn about using application data, creating background tasks, and working with Web services in Windows 8-style apps. Other skills you will learn include selecting user contacts in a Windows 8-style app, and securely storing and managing your identification information. You will also learn about network services for Windows 8-style apps, and background transfer.

Core development tasks
- Create a function that returns a promise
- Handle errors
- Chain promises
- Manage application data, file access, and Local/Roaming settings
- Network file transfers
- Work with encryption and certificates
- Manage background tasks
- Manage background data transfer
- Use Password Vault
- Manage contacts

Prerequisites
We assume that you’re already familiar with the concepts of programming and you've some experience of Visual Studio. For this Drill, you'll also need to be comfortable with JavaScript and HTML5 syntax. Although not essential, experience with Visual C# would also be useful. If you don’t have prior experience of creating Windows 8-style apps, we recommend looking at the Application Development Foundation drill for Windows 8.
Drill: Hardware & Multimedia

Audience
This Drill is for .NET developers who want to learn how to control the presentation, navigation and behavior of their Windows 8-style apps.

What you’ll learn
You will first learn about application lifecycle events and how to handle them. You will then learn about playing multimedia files in your apps, how to control the playing of them, and how to play them in the background.

Core development tasks
• use and handle application lifecycle events, including the launch, suspend and resume events
• play audio and video in your apps
• use the MediaElement to control multimedia
• enable the playing of background audio

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio. For this Drill, you’ll also need to be comfortable with JavaScript and HTML5 syntax. Although not essential, experience with Visual C# would also be useful. If you don’t have prior experience of creating Windows 8-style apps, we recommend looking at the Application Development Foundation drill for Windows 8.
Drill: Contracts

Audience
This Drill is for .NET developers who want to learn about how Window 8-style apps use contracts to declare the interactions that they support with other apps.

What you'll learn
You will first learn how to configure the search contract to respond to and support search queries. You will then learn about the sharing contract, and how to share text, images and URLs. Other skills you will learn include the use of the app to app picker contract, to provide shared access to files between apps. You will also learn about how stream images, audio, and video using the play to contract.

Core development tasks
- Configure a search contract
- Respond to a search query
- Add search suggestions and implement filters
- Handle selection of a search result item
- Share text, images, and URLs
- Setup a share target contract
- Handle the share operation outside the activation handler
- Implement long-running share support
- Set up File Open Picker contract
- Provide access to files
- Provide a save location
- Configure a play to contract
- Stream images, audio and video to a device on the network

Prerequisites
We assume that you're already familiar with the concepts of programming and you've some experience of Visual Studio. For this Drill, you'll also need to be comfortable with JavaScript and HTML5 syntax. Although not essential, experience with Visual C# would also be useful. If you don't have prior experience of creating Windows 8-style apps, we recommend looking at the Application Development Foundation drill for Windows 8.
Windows 8 App Development with C# and XAML

In this Developer Program, you will learn the fundamentals of Windows 8 app development using C# and XAML, with an emphasis on user experience elements and using Windows 8-style controls.

This Developer Program contains the following Drills:

- Windows 8 App Development Foundation
- Windows 8-Style UI Layouts and Surfaces
- Resource Management
- Contracts
Drill: Windows 8 App Development Foundation

This Drill will familiarize you with how to create a Windows 8-style app, and how to add controls and make them work. You will learn some of the basics of asynchronous programming in C#. You will also get some experience of the Windows Store.

Audience
This Drill is for .NET developers who want to learn how to create Windows 8-style apps.

What you'll learn
You will first learn how to create a Windows 8-style app using C#. You will then learn how to use the Canvas element, how to build app bar surfaces, and how to use the file picker. Other skills you will learn include how to add event handlers to controls, how to modify controls’ content, and how to fill your app with graphics. You will also learn about asynchronous programming and how to monetize an app using the Windows Store.

Core development tasks
- Create new projects using the correct templates
- Create live tiles and secondary tiles
- Add content and draw to the app window
- Add app bar commands
- Use FilePicker
- Handle control events
- Animate graphics
- Use asynchronous programming techniques
- Use trial conversion and in-app purchase in the Windows Store

Prerequisites
We assume that you're already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you'll also need to be comfortable with C# and XAML syntax.
Drill: Windows 8-Style UI Layouts and Surfaces

This Drill will familiarize you with how to use the available user interface surfaces and notifications. You will learn how to control layouts of supported views. You will also learn about navigating between pages, and how to create live tiles.

**Audience**
This Drill is for .NET developers who want to learn how to control the presentation, navigation and behavior of their Windows 8-style apps.

**What you'll learn**
You will first learn how to control the behavior of your app using the app bar and charms. Other skills you will learn include how provide suitable navigation for Windows 8-style apps, and how to reference and display external content. You will also learn how to use live tiles and badges to display app data on the start screen.

**Core development tasks**
- add content to the app window
- add app bar commands
- add settings using charms
- navigate between pages
- reference content in the application and link to external pages
- add live and secondary tiles
- update tiles

**Prerequisites**
We assume that you're already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you'll also need to be comfortable with C# and XAML syntax.

---

**Drill Details**

**Status**
Released

**Developer Program**
Windows 8 App Development with C# & XAML

**Number of Tasks**
3

**Duration**
- Research: 1.5 hours
- Coding: 1.5 hours

**Programming Language**
C#

**System Requirements**
Visual Studio 2012
Windows 8 Runtime

**Level**
3

**Drill Code**
- D0068B-CS
Drill: Resource Management

This Drill will familiarize you with resource management in Windows 8. You will learn about some of the more advanced aspects of asynchronous programming for C# Windows 8-style apps. You will also learn how to manage application data, and how to work with network services.

**Audience**
This Drill is for .NET developers who want to learn about resource management for Windows 8-style apps, how to manage application resources asynchronously, and application data across files and networks.

**What you’ll learn**
You will first learn about asynchronous programming in JavaScript, focusing on some of the more difficult topics. You will then learn about using application data, creating background tasks, and working with Web services in Windows 8-style apps. You will also learn about network services for Windows 8-style apps, and background transfer.

**Core development tasks**
- Create a function that returns a promise
- Handle errors
- Chain promises
- Manage application data, file access, and Local/Roaming settings
- Network file transfers
- Work with encryption and certificates
- Manage background tasks
- Manage background data transfer

**Prerequisites**
We assume that you’re already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you’ll also need to be comfortable with C# and XAML syntax.
Drill: Contracts

This Drill will familiarize you with Windows 8-style app contracts. Contracts define the requirements that apps must meet to participate in these Windows interactions. You will learn how to use the following contracts: search and sharing.

**Audience**
This Drill is for .NET developers who want to learn about how Windows 8-style apps use contracts to declare the interactions that they support with other apps.

**What you'll learn**
You will first learn how to configure the search contract to respond to and support search queries. You will then learn about the sharing contract, and how to share text, images and URLs.

**Core development tasks**
- Configure a search contract
- Respond to a search query
- Add search suggestions and implement filters
- Handle selection of a search result item
- Share text, images, and URLs
- Setup a share target contract
- Handle the share operation outside the activation handler
- Implement long-running share support

**Prerequisites**
We assume that you're already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you'll also need to be comfortable with C# and XAML syntax.
Windows Azure

This Developer Program contains the following Drills:

- Windows Azure Web Sites
- Windows Azure Virtual Machines
- Windows Azure Cloud Services Install
- Windows Azure Storage Service
- Windows Azure Caching
- Windows Azure Service Bus
- Windows Azure and Windows 8
- Advanced Cloud Services
- Windows Azure Usage Scenarios
- Windows Azure Advanced Storage Services
Drill: Windows Azure Web Sites

This Drill will familiarize you with how to make use of Windows Azure Web Sites, which simplifies and speeds up the deployment of Web sites to a highly scalable cloud environment. You will learn how to create, deploy and modify a Web site. You will also learn about Node.js applications and Git. And you'll get to use the Windows Azure Management Portal and Microsoft Team Foundation Service.

**Audience**
This Drill is for .NET developers who want take advantage of Windows Azure Web Sites to create, deploy and manage cloud-based Web sites.

**What you'll learn**
You will first learn how to use Windows Azure Web Sites to create a Web site in the Azure Management Portal, deploy an ASP.NET MVC4 Web site to it, and perform some basic maintenance on the site. You'll then look at Node.js and Git, creating a Node.js application, and configuring Git to deploy and manage the application. Other skills you will learn include using the Management Portal Image Gallery, Web site diagnostics, adding your applications to source control with Microsoft Team Foundation Service.

**Core development tasks**
- Create an ASP.NET MVC4 Web site using Visual Studio 2012
- Deploy the site and update it using Web Deploy
- Install GIT and create a new web site using the command line tools in GIT Bash
- Create a simple Node.js application
- Configure GIT publishing locally and in Azure
- Push the web site using GIT, modify, and republish
- Turn on diagnostics and review the log files
- Scale a site
- Connect a Windows Azure Web Site to a TFS Project and Visual Studio 2012 to a TFS Repository

**Prerequisites**
We assume that you're already familiar with the concepts of programming and you've some knowledge of how Windows Azure works. For this Drill, you'll also need to be comfortable with C# syntax.
Drill: Windows Azure Virtual Machines

This Drill will familiarize you with how to create various types of virtual machine using an array of tools available to you in Windows Azure. You will learn how to create Windows Server, Linux and SQL Server VMs. And you’ll get to use the Windows Azure Management Portal, PowerShell cmdlets, Remote Desktop and tools for Mac and Linux.

**Audience**
This Drill is for .NET developers who want to use Windows Azure to work with virtual machines in a variety of formats.

**What you'll learn**
You will first learn how to use both the Windows Azure Management Portal and PowerShell cmdlets to create Windows Server VMs. You’ll then look at Linux VMs, creating them using both the Azure Management Portal and the command line tools for Mac and Linux. You’ll also create an Endpoint for a Linux VM. Other skills you will learn include how to create a SQL Server Virtual Machine using the Management Portal, and how to use Remote Desktop to configure it to be accessible across the internet.

**Core development tasks**
- Understand the capabilities of automating the deployment and management of virtual machines in Windows Azure
- Create a Windows Server VM and attach a disk using the Management Portal
- View the disk and capture the image
- Use the Windows Azure IaaS platform to provision a Linux-based VM in the cloud and manage it remotely.
- Use Windows Azure command-line tools for Linux
- Work with SQL Server VMs in Windows Azure

**Prerequisites**
We assume that you’re already familiar with the concepts of programming and you’ve some knowledge of how Windows Azure works. For this Drill, you'll also need to be comfortable with C# syntax.
Drill: Windows Azure Cloud Services

This Drill will familiarize you with how to make use of Windows Azure Cloud Services, which allows you to quickly deploy and manage your applications in a modern cloud architecture. You will learn how to create, deploy and modify Web and Worker roles. You will also learn how to monitor your cloud services and make appropriate adjustments.

Audience
This Drill is for .NET developers who want take advantage of Windows Azure Cloud Services to create, deploy and manage cloud-based applications.

What you'll learn
You will first learn how to use Windows Azure Cloud Services, and in particular how to use Web and Worker roles to deploy an application to a newly created Azure Cloud Service. You'll then look at cloud service performance, and learn what's available.

Core development tasks
• Create an Asp.Net MVC4.0 Web Role in Visual Studio 2012
• Deploy Cloud Services to the Azure emulator
• Create a cloud service using Quick Create on the Management Portal
• Upload a management certificate for a cloud service
• Deploy a cloud service from Visual Studio 2012
• Create a worker role in Visual Studio 2012
• Deploy to file system
• Deploy to Staging slot using the Portal
• Move to Production slot using the Portal
• Configure monitoring for a worker role
• Set up monitoring to use the storage emulator
• Add metrics

Prerequisites
We assume that you're already familiar with the concepts of programming and you've some knowledge of how Windows Azure works. For this Drill, you'll also need to be comfortable with C# syntax.

Drill Details
Status
Released
Developer Program
Windows Azure
Number of Tasks
3
Duration
Research: 1.5 hours
Coding: 1.5 hours
Programming Language
C#
System Requirements
Visual Studio 2012
Access to Windows Azure
Level
3
Drill Code
• D0069C-CS
Drill: Windows Azure Storage Services

The Windows Azure Blob Storage Service allows extremely large amounts of data to be hosted in and retrieved from the cloud. We'll look at uploading, listing, downloading, and deleting blobs.

Audience
This Drill is for .NET developers who want to avail of cloud storage of large amounts of data using Windows Azure.

What you'll learn
You will learn how to perform common storage scenarios using the Windows Azure Blob storage service. The scenarios include uploading, listing, downloading, and deleting blobs. The samples are written in C# and use the .NET API. You will also learn about Windows Azure Table Services, and how to create, delete and update a table.

Core development tasks
- Create a blob container
- Upload blob into container
- List blobs
- Download a blob
- Delete a blob
- Create a table
- Add an entity to a table
- Retrieve all entities in a partition
- Update an entity
- Delete an entity

Prerequisites
We assume that you're already familiar with the concepts of programming and you've some knowledge of how Windows Azure works. For this Drill, you'll also need to be comfortable with C# syntax.
Drill: Windows Azure Caching

The Windows Azure Caching service provides a cache in the cloud that can be used in any applications or services that could benefit from caching. We'll look at creating a cache, and adding and retrieving cache data.

**Audience**
This Drill is for .NET developers who want to create applications that can benefit from caching data in the cloud using Windows Azure.

**What you'll learn**
You will learn how to create a cache programmatically, and add and retrieve cache data. The samples are written in C# and use the .NET API.

**Core development tasks**
- Enable cache service
- Use cache service to store session state
- Store data retrieved from database in cache

**Prerequisites**
We assume that you're already familiar with the concepts of programming and you've some knowledge of how Windows Azure works. For this Drill, you'll also need to be comfortable with C# syntax.
Drill: Windows Azure Service Bus

The Windows Azure service bus provides connectivity options for WCF and other service endpoints. We’ll look at setting up and accessing services.

Audience
This Drill is for .NET developers who want to become familiar with the Windows Azure Service Bus secure hosted communication infrastructure.

What you'll learn
In this drill you’ll learn how to set up a service bus namespace, retrieve the authentication key from the management portal, and configure the WCF endpoints to connect to the relay. You’ll also learn how to use the Queue Storage Service in a Windows Azure application.

Core development tasks
• Configure the WCF service to expose a service bus endpoint in config file
• Configure an Azure web role to consume the service
• Write to a queue
• Read from a queue
• Inspect message properties

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some knowledge of how Windows Azure works. For this Drill, you’ll also need to be comfortable with C# syntax.

Drill Details
Status
Released
Developer Program
Windows Azure
Number of Tasks
2
Duration
Research: 1 hour
Coding: 1 hour
Programming Language
C#
System Requirements
Visual Studio 2012
Access to Windows Azure
C#
Level
2
Drill Code
• D0069F-CS
Drill: Windows Azure & Windows 8

This Drill will familiarize you with how to make use of Azure’s cloud services from within your Windows 8 apps. You will learn how to create, publish and consume a Web API service. You will also learn about the Windows Push Notification Services, and how to send toast notifications.

Audience
This Drill is for .NET developers who want to connect their Windows Store apps with the cloud services provided by Windows Azure.

What you'll learn
You will first learn how to create an ASP.NET MVC 4 Web API service and publish it to Windows Azure Web Sites. You’ll then consume the Web API service in a Windows 8 app. Other skills you will learn include how to configure the Windows Push Notification Services (WNS) in your app to send toast notifications from your service to all of the registered clients.

Core development tasks
- Create a Web API Service
- Publish the service to Azure Web Sites
- Create a simple Windows Store app to consume the service
- Bind the application to the data of the Web API service
- Request WNS Credentials and update the Web.config
- Set up notification app server
- Configure a Windows Store app to receive notifications

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some knowledge of how Windows Azure works. For this Drill, you’ll also need to be comfortable with C# and XAML syntax. If you don’t have prior experience of creating Windows 8-style apps, we recommend looking at the Application Development Foundation drill for Windows 8.
Drill: Advanced Cloud Services

This Drill will familiarize you with how to make use of some of the advanced features of the Windows Azure Cloud Services, which allows you to quickly deploy and manage your applications in a modern cloud architecture.

Audience
This Drill is for .NET developers who want take advantage of Windows Azure Cloud Services to create, deploy and manage cloud-based applications.

What you'll learn
You will learn how to control the emulator, package cloud services, and deploy them to the cloud. You will also learn how to configure startup tasks for a Windows Azure Web role, and how to use Windows Azure Service Management REST API to manage your service deployments.

Core development tasks
- Create a Web API Service
- Publish the service to Azure Web Sites
- Create a simple Windows Store app to consume the service
- Bind the application to the data of the Web API service
- Request WNS Credentials and update the Web.config
- Set up notification app server

Prerequisites
We assume that you're already familiar with the concepts of programming and you've some knowledge of how Windows Azure works. For this Drill, you'll also need to be comfortable with C# and XAML syntax.
Drill: Windows Azure Usage Scenarios

To achieve the scalability and reliability that cloud applications promise, you need to employ a multi-tier architecture. In this drill you will learn how to create a multi-tier application in Windows Azure using multiple roles and message-based communication between them.

Audience
This Drill is for .NET developers who want to create applications that can benefit from a multi-tier architecture in the cloud using Windows Azure.

What you’ll learn
You will learn how to create a multi-tier application in Windows Azure using Web and Worker roles, and how to communicate between tiers using Service Bus Queues.

Prerequisites
We assume that you're already familiar with the concepts of programming and you've some knowledge of how Windows Azure works. For this Drill, you'll also need to be comfortable with C#.
Drill: Windows Azure Advanced Storage Service

The Windows Azure Storage Service allows extremely large amounts of data to be hosted in and retrieved from the cloud. We'll look at some more advanced aspects of the Storage Service, including the REST API, concurrency, and data partitioning. We'll also look at performance enhancement using server side paging, batch operation and details of query performance.

**Audience**
This Drill is for .NET developers who want to avail of cloud storage of large amounts of data using Windows Azure.

**What you'll learn**
You will learn how to perform common CRUD operations using the rest API, and how to handle optimistic concurrency. You'll look at server-side paging and how to use continuation tokens, batch operations and query filters. You'll also get experience of improving the performance of domain queries, and subscribing to messages in topics using filtered subscriptions.

**Core development tasks**
- Create, read, update and delete operations with the REST API
- Optimistic concurrency
- The Etag property
- Server side paging
- Continuation tokens
- Batch operations
- Custom query filters
- Domain queries
- Service Bus topics and subscriptions

**Prerequisites**
We assume that you're already familiar with the concepts of programming and Azure, and that you’ve previously completed the Windows Azure Storage Service drill. For this drill, you’ll also need to be comfortable with C#.
Windows Phone 8 Development

In this Developer Program, you will learn the fundamentals of Windows Phone 8 app development using C# and XAML. There is an emphasis on user experience elements and using Windows Phone controls. You will also learn about the hardware and media that Windows Phone 8 devices offer, and how to manage resources.

This Developer Program contains the following Drills:

- Introduction to XAML based Windows Phone 8 Development
- Windows Phone 8 User Experience
- Windows Phone 8 Features
- Windows Phone 8 Launchers and Choosers
- Accessing Local and Remote Data
- Launching, Resuming and Multitasking
- Windows Phone 8 and Windows 8 Integration
Drill: Introduction to XAML based Windows Phone 8 Development

This Drill will familiarize you with how to create a Windows Phone 8 app, and how to add controls and make them work. You will learn some of the basics of UI development and navigation in a Phone 8 app. You will also get some experience of the Windows Store.

**Audience**
This Drill is for .NET developers who want to learn how to create Windows Phone 8 apps.

**What you'll learn**
You will first learn how to create a Windows Phone 8 app using C#, and set some basic interface attributes. You will then learn how to build a basic application structure using the Windows Phone Panorama App template, how to create a user interface for Windows Phone 8 using XAML. Other skills you will learn include how to implement navigation between pages in a Windows Phone 8 app, how to handle application lifecycle events to provide continuity when switching between apps. You will also learn about monetization in your apps using in-app purchasing from the Windows Phone Store.

**Core development tasks**
- Create new projects using the correct templates
- Configure tiles
- Use controls, including the grid, label, and button controls
- Manage User interface resources, styles, and data binding
- Use the app hub
- Manage the UI using the Panorama control
- Add an App bar
- Use the NavigationService, frames, pages, and the Back button
- Manage application states
- Use In-app purchasing

**Prerequisites**
We assume that you're already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you'll also need to be comfortable with C# and XAML syntax.

**Drill Details**

<table>
<thead>
<tr>
<th>Status</th>
<th>Released</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Program</td>
<td>Windows Phone 8 Development</td>
</tr>
<tr>
<td>Number of Tasks</td>
<td>6</td>
</tr>
<tr>
<td>Duration</td>
<td>Research: 3 hours Coding: 3 hours</td>
</tr>
<tr>
<td>Programming Language</td>
<td>C#</td>
</tr>
<tr>
<td>System Requirements</td>
<td>Visual Studio 2012 C# Windows Phone 8 SDK</td>
</tr>
<tr>
<td>Level</td>
<td>3</td>
</tr>
<tr>
<td>Drill Code</td>
<td>D0072A-CS</td>
</tr>
</tbody>
</table>
Drill: Windows Phone 8 User Experience

This Drill will familiarize you with how to use notifications in Windows Phone 8, and how to integrate your apps with the user interface elements, such as the app bar and system tray, that the system provides. You will learn some of the basics of how to display HTML content, and how to control the playback of media in a Phone 8 app. You will also get some experience binding control properties to data.

**Audience**
This Drill is for .NET developers who want to learn how to create Windows Phone 8 apps.

**What you'll learn**
You will first learn how to use a selection of built-in notifications in Windows Phone 8, including the Toast, Tile, Alarm, Lockscreen and Push notifications, and see where they can be applied. You will then learn the basics of Windows Phone 8 user interface features such as the app bar, the system tray, and the popup control, and how to handle different device resolutions. Other skills you will learn include how to display and control a content control. You will also learn about binding control properties to data, and look in particular at the LongListSelector control.

**Core development tasks**
- Use Local and Push notifications
- View notifications in Toasts, Tiles, Alarms and the lock screen
- Use the app bar
- Use the status bar
- Support multiple resolutions
- Use progress bars
- Show information in popups
- Display images
- Use the MediaElement control
- Use the WebBrowser control

**Prerequisites**
We assume that you’re already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you’ll also need to be comfortable with C# and XAML syntax.

---

**Drill Details**

**Status**
Released

**Developer Program**
Windows Phone 8 Development

**Number of Tasks**
5

**Duration**
Research: 2.5 hours
Coding: 2.5 hours

**Programming Language**
C#

**System Requirements**
Visual Studio 2012
C#
Windows Phone 8 SDK

**Level**
3

**Drill Code**
- D007Z-CS
Drill: Windows Phone 8 Features

This Drill will familiarize you with the orientation of a Windows Phone 8 device; how to manage landscape and portrait modes, and how to use the accelerometer. You will learn some of the basics of how to work with the geolocation API, and how to create lens apps. You will also get some experience of using voice commands, and integrating synthesized speech.

**Audience**
This Drill is for .NET developers who want to learn how to create Windows Phone 8 apps.

**What you'll learn**
You will first learn how to use determine orientation of a Windows Phone 8 device and update the UI accordingly, and you’ll use the built-in accelerometer to detect the forces of gravity and movement of the phone. You will then learn how to how to get the device’s current location, calculate the distance between two points on the map, and display a pushpin for any location on the map, all using the Geolocation API. Other skills you will learn include how to create a rich media lens, and how to process in real time image frames coming from the camera. You will also learn about using voice commands to launch your app, interacting with your app using speech, generating synthesized speech in your app, and how to handle speech recognition errors.

**Core development tasks**
- Respond to orientation changes and set supported orientations
- Initialize geolocation
- Get the current location
- Set up a watcher
- Use the accelerometer
- Integrate a lens app with the camera
- Create a rich media lens
- Use voice commands and speech recognition
- Use Text To Speech
- Handle speech errors

**Prerequisites**
We assume that you're already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you'll also need to be comfortable with C# and XAML syntax.
Drill: Windows Phone 8 Launchers and Choosers

This drill will introduce you to Launchers and Choosers, which are APIs that can be used to bring a user’s content into your app. Making use of Launchers and Choosers helps to ensure that your app looks similar to other Windows Phone 8 apps, and works in a way that is familiar to users.

Audience
This Drill is for .NET developers who want to learn how to create Windows Phone 8 apps.

What you'll learn
In this challenge, you will learn to use a Chooser to request content form a user, how to allow your app to be launched from content it supports, and how to manage the application state between these operations. You'll also learn how to work with the Maps Launchers to connect an app to the geolocation and maps features of a device.

Core development tasks
• Use the camera capture task
• Use the share media task
• Save and restore application state when using chooser
• Use the maps task
• Use the Maps directions task
• Use the Maps downloader task
• Use the Maps updater task

Prerequisites
We assume that you're already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you'll also need to be comfortable with C# and XAML syntax.

Drill Details

<table>
<thead>
<tr>
<th>Status</th>
<th>Released</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Program</td>
<td>Windows Phone 8 Development</td>
</tr>
<tr>
<td>Number of Tasks</td>
<td>2</td>
</tr>
<tr>
<td>Duration</td>
<td>Research: 1 hour, Coding: 1 hour</td>
</tr>
<tr>
<td>Programming Language</td>
<td>C#</td>
</tr>
<tr>
<td>System Requirements</td>
<td>Visual Studio 2012, C# Windows Phone 8 SDK</td>
</tr>
<tr>
<td>Level</td>
<td>3</td>
</tr>
<tr>
<td>Drill Code</td>
<td>D0072D-CS</td>
</tr>
</tbody>
</table>

Drill: Accessing Local and Remote Data

In this challenge, you'll learn how to use Isolated Storage, and XML and JSON serializers. You'll also learn how to access a Web service and get network information for a specific Windows Phone device. And you will learn how to display an external web page and deserialize JSON data.

**Audience**
This Drill is for .NET developers who want to learn how to create Windows Phone 8 apps.

**What you'll learn**
You will first learn how to use both name-value pairs and files for isolated storage. You'll also use XML and JSON variations on IsolatedStorageFile. You will then learn about Web services and how the Windows Phone 8 SDK provides classes to work with remote Web services. You'll use classes that allow an app to get information about the network status of the device it is running on.

**Core development tasks**
- Use XML serialization
- Use JSON serialization
- Store small data in Isolated Storage
- Store large data using a Storage File
- Get network information
- Call a JSON service
- Parse JSON
- Use an ODATA service
- Embed a WebBrowser

**Prerequisites**
We assume that you're already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you'll also need to be comfortable with C# and XAML syntax.
Drill: Launching, Resuming and Multitasking

In this drill, you'll learn about the lifecycle of a Windows Phone app and how your app should handle different lifecycle events. You'll also learn how to enable fast app resume for Windows Phone 8 apps. You'll learn how to work with the background transfer service, and you'll learn how to create a location-tracking app that runs in the background and displays locations on the map control.

**Audience**
This Drill is for .NET developers who want to learn how to create Windows Phone 8 apps.

**What you'll learn**
You will walk through how to use a set of events and related APIs to allow your app to handle different lifecycle events. You'll also see how to use the Windows Phone 8 background transfer APIs to queue, execute and monitor multiple uploads and downloads over HTTP while users continue to use the app features in the foreground. And you'll become familiar with the location API to build apps capable of tracking location while running in the background.

**Core development tasks**
- Handle page state
- Handle app state
- Use idle detection
- Implement fast app resume
- Create a background transfer request
- Monitor background transfer requests
- Get a request's status and display detailed status
- Display a map with marked locations
- Track locations for a moving device
- Plot a trail of location on a map

**Prerequisites**
We assume that you're already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you'll also need to be comfortable with C# and XAML syntax.

---

**Drill Details**

**Status**
Released

**Developer Program**
Windows Phone 8 Development

**Number of Tasks**
3

**Duration**
Research: 1.5 hours
Coding: 1.5 hours

**Programming Language**
C#

**System Requirements**
Visual Studio 2012
C#
Windows Phone 8 SDK

**Level**
3

**Drill Code**
- D0072F-CS
Drill: Windows Phone 8 and Windows 8 Integration

In this drill, you’ll learn about the creating apps for multiple devices, how you can make the development process as efficient as possible by sharing functionality, and how you can manage the fundamental differences between the two platforms.

Audience
This Drill is for .NET developers who want to learn how to create Windows Phone 8 apps.

What you'll learn
You’ll discover and apply some of the techniques available for sharing code between Windows Store and Windows Phone apps, which includes sharing libraries and source code directly. In this coding challenge, you’ll learn how to handle platform differences, and, using visual states, you will change the appearance of controls when, for example, validation fails.

Core development tasks
• Reuse code using portable class libraries
• Share code with add as a link
• Share using WinRT components
• Use Conditional compilation
• Handle differences between the platforms: Markup extensions, Manipulation events, Binding modes, Data validation, Visual states

Prerequisites
We assume that you’re already familiar with the concepts of programming and you have some experience of Visual Studio. For this Drill, you’ll also need to be comfortable with C# and XAML syntax.
Modern Web Programming with HTML5

This Developer Program contains the following Drill:

• HTML5-based Applications
Drill: HTML5-based Applications

**Audience**
This Drill is for Web developers who want to learn how to create Web applications that take advantage of HTML5 features.

**What you'll learn**
You will learn how to create pages using new HTML5 elements and gain experience in using new element attributes. You will also learn how to use the Canvas element to display a graph. Other HTML5 skills you will learn include how to manipulate browser history, to work with video and audio elements, and to use local and session storage. You will also learn how to run a Web application offline and how to adapt Web applications to mobile devices.

**Core development tasks**
- use new HTML5 elements and attributes
- display a graph using JavaScript
- use the canvas element
- manipulate browser history
- work with video and audio elements
- use local and session storage
- run a Web application offline
- adapt a Web application to mobile devices

**Prerequisites**
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio 2008/2010. For this Drill, you’ll also need to be comfortable with JavaScript and HTML syntax. Although not essential, experience with Visual C# would also be useful.

**Drill Details**

<table>
<thead>
<tr>
<th>Status</th>
<th>Released</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Program</td>
<td>Modern Web Programming with HTML5</td>
</tr>
<tr>
<td>Number of Tasks</td>
<td>6</td>
</tr>
<tr>
<td>Duration</td>
<td>3 hours</td>
</tr>
<tr>
<td>Programming Language</td>
<td>C#</td>
</tr>
<tr>
<td>System Requirements</td>
<td>An HTML5 compatible browser</td>
</tr>
<tr>
<td>Level</td>
<td>2</td>
</tr>
<tr>
<td>Drill Code</td>
<td>D0066A-CS</td>
</tr>
</tbody>
</table>
Threading in Visual C#

This Developer Program contains the following Drill:

- .NET 4.0 Concurrency
Drill: .NET 4.0 Concurrency

Audience
This Drill is for developers who want to learn how to use concurrency in .NET 4.0.

What you'll learn
You'll learn how to create and use Tasks to run code asynchronously, how to enumerate collections concurrently with the Parallel class and how to use PLINQ. You'll also learn how to execute code lazily as well as cancel tasks you've started. Finally, you'll learn how to use the concurrent collections and dictionary .NET provides.

Core development tasks
- use the Task class to run tasks in their own threads and organize their cooperation
- use the Parallel class to execute loops concurrently
- execute parallelized LINQ statements
- use lazy evaluation in parallel computing
- cancel tasks and use speculative processing in concurrent computation
- use collections and dictionaries in multithreaded applications

Prerequisites
We assume that you’re already familiar with programming concepts and that you’ve used Visual Studio. For this particular drill, you'll also need to be comfortable with Visual C# syntax. Although not essential, some familiarity with threading fundamentals would be useful.
Performance in the .NET Framework

This Developer Program contains the following Drill:

• Improving Performance in .NET Framework Applications
Drill: Improving Performance in .NET Framework Applications

Audience
This Drill is for experienced developers who want to learn key techniques required to optimize the performance of applications in the .NET Framework.

What you'll learn
You'll learn how to measure application performance, implement a custom generic cache, employ task parallelism, and execute object pooling. You'll also learn how to use lazy evaluation and deferred execution, as well as AJAX and CSS Sprites.

Core development tasks
• measure an application's performance using managed performance counters and the Stopwatch class
• improve the performance of a Windows Forms application using a custom generic cache
• split a long computation into small sub tasks and distribute them across different processors to be processed concurrently
• use an object pool to reuse objects that are expensive to create or frequently created
• postpone the execution of expensive calculations until they are absolutely required
• improve the performance of ASP.NET applications by using AJAX, CSS Sprites, and caching.

Prerequisites
We assume that you're already familiar with programming concepts and that you've used Visual Studio. For this particular drill, you'll also need to be comfortable with Visual C# syntax, Windows Forms programming, and ASP.NET web development.
Threading in C#/VB using Visual Studio

This Developer Program contains the following Drills:

- Threading in the .NET Framework
- Concurrency Patterns
- .NET 4.0 Concurrency
Drill: Threading in the .NET Framework

Audience
This Drill is for developers who want to learn basic techniques required to work with threads in the .NET Framework.

What you'll learn
You'll learn how to use the .NET thread pool, the BackgroundWorker component, thread safe programming techniques, and asynchronous delegates. You'll also learn how to perform thread synchronization with wait handles, as well as how to create and control threads manually.

Core development tasks
- use threads efficiently with the .NET thread pool
- use the BackgroundWorker component to perform a long search operation while reporting its progress
- use the Thread class to manually create foreground and background threads, and to synchronize execution between two manual threads
- use synchronization primitives to coordinate dependent activities across threads
- synchronize critical regions with the Monitor class
- call synchronous methods in an asynchronous manner via delegates.

Prerequisites
We assume that you’re already familiar with programming concepts and that you’ve used Visual Studio. For this particular drill, you’ll also need to be comfortable with Visual C# syntax and Windows Forms programming. Although not essential, some familiarity with the concept of threading would also be useful.
Drill: Concurrency Patterns

Audience
This Drill is for developers who want to learn advanced techniques required to work with threads in the .NET Framework.

What you'll learn
You'll learn about the Event-based Asynchronous Pattern, the Countdown Latch, and the Producer/Consumer relationship. You'll also learn how to parallelize data by using Loop Tiling, avoid deadlocks with Lock Leveling, and work with immutable types.

Core development tasks
• add asynchronous functionality to your own class to start, cancel, and track the progress of asynchronous tasks
• manage the execution of multiple threads using the countdown latch pattern
• complete a multi-threaded application that represents a producer/consumer relationship using the Monitor class
• parallelize the processing of data with loop tiling, while applying Map and Reduce functions over a set of input data
• acquire multiple locks in a fixed, global order to avoid deadlocks
• create an immutable type and display an immutable snapshot of a collection in the UI, while also enabling multiple threads to modify the collection.

Prerequisites
We assume that you’re already familiar with programming concepts and that you’ve used Visual Studio. For this particular drill, you’ll also need to be comfortable with Visual C# syntax and Windows Forms programming. Although not essential, some familiarity with threading fundamentals would also be useful.
Drill: .NET 4.0 Concurrency

Audience
This Drill is for developers who want to learn how to use concurrency in .NET 4.0.

What you'll learn
You'll learn how to create and use Tasks to run code asynchronously, how to enumerate collections concurrently with the Parallel class and how to use PLINQ. You'll also learn how to execute code lazily as well as cancel tasks you've started. Finally, you'll learn how to use the concurrent collections and dictionary .NET provides.

Core development tasks
• use the Task class to run tasks in their own threads and organize their cooperation
• use the Parallel class to execute loops concurrently
• execute parallelized LINQ statements
• use lazy evaluation in parallel computing
• cancel tasks and use speculative processing in concurrent computation
• use collections and dictionaries in multithreaded applications

Prerequisites
We assume that you're already familiar with programming concepts and that you've used Visual Studio. For this particular drill, you'll also need to be comfortable with Visual C# syntax. Although not essential, some familiarity with threading fundamentals would be useful.
ASP.NET MVC using Visual Studio

This Developer Program contains the following Drill:

• ASP.NET MVC Fundamentals
Drill: ASP.NET MVC Fundamentals

**Audience**
This Drill is for intermediate developers who want to learn how to create Web applications with ASP.NET MVC.

**What you'll learn**
You'll learn how to work effectively with controllers, actions, and views, and how to use MVC's built-in scaffolding feature, as well as how to validate input data. You'll also learn how to use filters and AJAX with MVC applications.

**Core development tasks**
- create an MVC application
- pass strongly-typed data from controller actions to views
- create actions that only run in response to specific HTTP request types (such as GET and POST)
- work with Select lists in view pages and actions, and reuse portions of the UI
- use the built-in scaffolding feature of ASP.NET MVC to quickly create views to manipulate
- the data in an application
- update a domain model using view data and validate the data against business logic
- determine whether validation has been successful and vary output accordingly
- add filters to controller classes to create extra functionality for an existing ASP.NET MVC
- application with minimal coding
- enable requests using AJAX in MVC and distinguish between normal and AJAX requests
- measure an application's performance using managed performance counters and the Stopwatch class

**Prerequisites**
We assume that you're already familiar with programming concepts and that you've used Visual Studio. For this drill, you'll also need to be comfortable with Visual C# syntax, basic Web development, and HTML fundamentals. Although not essential, it would be useful to have some familiarity with the HTTP model, the concepts behind AJAX, and the Model View Controller pattern.
JavaScript Development with jQuery

This Developer Program contains the following Drill:

- Developing with jQuery
Drill: Developing with jQuery

Audience
This Drill is for developers of all career levels looking to expand existing Web development skills.

What you'll learn
Our hands-on challenges will provide a practical understanding of the rapid and concise jQuery library. Topics include selectors, traversing, manipulation, events, animation effects, CSS, custom plugins, and AJAX.

Core development tasks
- using jQuery selectors to find elements on a Web page and jQuery's effects AP to create an animated user interface.
- traversing and manipulating DOM elements
- handling events raised from DOM elements
- modify the style of DOM elements using CSS
- using AJAX features in jQuery to make HTTP GET and POST requests
- authoring a custom plugin to add extra methods and functionality to jQuery applications.

Prerequisites
We assume that you're familiar with programming concepts, Web development, and Visual Studio. For this particular drill, you'll also need to be comfortable with JavaScript programming and HTML syntax. Although not essential, some familiarity with AJAX and CSS would also be useful.
This Developer Program contains the following Drills:

- New Features in C# 4.0
- ASP.NET MVC Fundamentals
- ADO.NET Entity Framework
Drill: New Features in C# 4.0

Audience
This Drill is intended for developers with experience in .NET who want to learn how to use the new features of .NET Framework 4.0 to enhance their programming skill set.

What you'll learn
You will learn how to perform dynamic programming, use named and optional arguments, and work with the enhanced interoperability in COM. You will also learn how to introduce variance into your generic interfaces and delegates.

Core development tasks
- use the dynamic keyword as a data type that supports runtime lookup of its members
- use optional parameters for constructors and methods by assigning default values to input arguments
- explicitly name an argument being passing to a method, rather than identifying it by argument position
- work with the enhanced COM Interop features in C# 4.0
- dynamically import COM APIs and deploy without Primary Interop Assemblies (PIA)
- skip the passing of optional parameters when making calls to COM objects
- omit the ref keyword when calling a method on a COM object that expects parameters to be passed by reference
- use the built-in .NET interfaces that have been made variant in .NET Framework 4.0
- make generic interfaces and delegates covariant.

Prerequisites
We assume that you're already familiar with programming concepts and that you've used Visual Studio. For this Drill, you'll also need to be comfortable with Visual C# syntax, basic Web development, and HTML fundamentals. It would also be useful to have some familiarity with previous versions of the .NET Framework.
Drill: ASP.NET MVC Fundamentals

**Audience**
This Drill is intended for intermediate developers who want to learn how to create Web applications with ASP.NET MVC.

**What you'll learn**
You'll learn how to work effectively with controllers, actions, and views, and how to use MVC's built-in scaffolding feature, as well as how to validate input data. You'll also learn how to use filters and AJAX with MVC applications.

**Core development tasks**
- create an MVC application
- pass strongly-typed data from controller actions to views
- create actions that only run in response to specific HTTP request types (such as GET and POST)
- work with Select lists in view pages and actions, and reuse portions of the UI
- use the built-in scaffolding feature of ASP.NET MVC to quickly create views to manipulate the data in an application
- update a domain model using view data and validate the data against business logic
- determine whether validation has been successful and vary output accordingly
- add filters to controller classes to create extra functionality for an existing ASP.NET MVC application with minimal coding
- enable requests using AJAX in MVC and distinguish between normal and AJAX requests

**Prerequisites**
We assume that you're already familiar with programming concepts and that you've used Visual Studio. For this drill, you'll also need to be comfortable with Visual C# syntax, basic Web development, and HTML fundamentals. Although not essential, it would be useful to have some familiarity with the HTTP model, the concepts behind AJAX, and the Model View Controller pattern.
Drill: ADO.NET Entity Framework

**Audience**
This Drill is for developers using Visual Studio 2010 who want to start using the ADO.NET Entity Framework to simplify data access in their applications.

**What you'll learn**
This Drill will introduce you to the ADO.NET Entity Framework, an object-relational mapper from Microsoft. You'll learn how to create an entity model and use it to query and modify data from your database. You will also learn how to adapt your model to use stored procedures, create your own entity classes and optimize performance.

**Core development tasks**
- expose an underlying database schema through an Entity Framework data model
- query an entity data model generated from an existing database
- make changes to the data in a database
- create, update, insert, and delete entities using existing stored procedures
- replace Entity Framework-generated entity classes with your own Plain Old CLR Objects (POCOs)
- create a custom ObjectContext derived class
- apply optimization techniques to LINQ to Entities queries to improve their performance

**Prerequisites**
We assume that you're already familiar with programming concepts and that you've used Visual Studio. For this drill, you'll also need to be comfortable with Visual C# syntax and basic database concepts.
Silverlight 4.0 Programming using Visual C#

This Developer Program contains the following Drill:

- Silverlight 4.0 Fundamentals
Drill: Silverlight 4.0 Fundamentals

**Audience**  
This Drill is for developers using Visual Studio 2010 who want to start using Silverlight 4 to create rich Internet and desktop-based applications.

**What you'll learn**  
You'll learn how to create a new Silverlight 4 project, how to layout and style your applications, and how to use video and other media elements. You'll also learn how to bind your data to Silverlight controls, how to display data using the DataGrid, and how to make your apps run outside the browser.

**Core development tasks**  
- create new Silverlight 4 projects
- apply layout and styling techniques to modify the visual appearance of your applications
- display video in your applications
- bind the data in your applications to Silverlight controls
- display data in a grid
- support drag and drop techniques
- run your applications outside the browser environment

**Prerequisites**  
We assume that you're already familiar with programming concepts and that you've used Visual Studio. For this Drill, you'll also need to be comfortable with Visual C# syntax.

**Drill Details**  
- **Status**: Released
- **Developer Program**: Silverlight 4.0 Programming using Visual C#
- **Number of Tasks**: 6
- **Duration**: 3 hours
- **Programming Language**: C#
- **Level**: 3
- **Drill Code**: D0046E-CS
Enterprise Library 5.0 using Visual C# 2008/2010

This Developer Program contains the following Drill:

- Enterprise Library 5.0
Drill: Enterprise Library 5.0

Audience
This Drill is for developers who want to learn how to use the Data Access, Validation, Exception Handling, Unity, and Caching Application Blocks of the Enterprise Library.

What you’ll learn
You will learn how to use several Enterprise Library application blocks to address common development challenges. The skills you will learn include using and configuring the Data Access Application Block to access a database, creating object validation with the Validation Application Block, throwing, catching, and logging exceptions with the Exception Handling Application Block, building scalable applications with dependency injection (DI), and working with the Caching Application Block.

Core development tasks
- configuring application blocks with the Enterprise Library Configuration Console
- using the Data Access Application Block to retrieve information from a database
- adding Enterprise Library references
- configuring the application to work with a database
- defining a mapper to easily transform a dataset into a business object
- executing simple queries and stored procedures
- configuring exception behavior, modifying and wrapping exceptions using configuration file setting validation rules in the app.config and code
- using built-in validators, setting up self-validation, changing validation messages in configuration files and in application resources
- displaying the results of a validation process
- defining the validation rules on a property or type
- assigning the validation method to controls events in XAML
- resolving the error messages in behind-code
- completing the validation processing in a WCF service
- enabling the use of a validator in WCF service attributes.
- configuring the service interface to validate the input parameter of a service method
- reading, writing and deleting cache content
- creating cache with the content in the persistent storage
- setting up the design-time and the run-time dependencies between objects
- resolving objects using a constructor and property injection

Prerequisites
We assume that you’re already familiar with programming concepts and that you’ve used Visual Studio. For this Drill, you’ll also need to be comfortable with Visual C# syntax, XML syntax, SQL syntax, as well as SQL databases (Express or Server). Although not essential, experience with individual application blocks or previous releases of Enterprise Library would also be useful.
ASP.NET Web Development with Visual Studio

This Developer Program contains the following Drills:

- Working with ASP.NET 2.0 Web Forms
- State Management and Page navigation
- Exception Handling and User Input Validation
- Rich Controls and Services
- Master Pages and Site Navigation
- Personalization and Themes
- Web Parts
- Creating ASP.NET 2.0 Controls
- Reading and Writing XML Data
- Consuming and Manipulating Data with ADO.NET
- Data Source Controls and Data Binding
- Using Data-bound Controls
- Membership, Login Controls and Forms Authentication
- Authorization and Roles
- Web Application Attacks
- Securing and Monitoring ASP.NET Applications
- Styling Websites with Cascading Stylesheets (CSS)

The following books cover the recommended prerequisites for this particular Developer Program:

- Programming Visual Basic by Jesse Liberty, published by O'Reilly Media, ISBN 0596009496
Drill: Working with ASP.NET 2.0 Web Forms

Overview
This Drill lets you create and configure ASP.NET 2.0’s web server controls and work on techniques involving data binding, dynamic property manipulation, accessibility and dynamic control creation.

Audience
This Drill is for developers who want to be able to understand and work comfortably with ASP.NET web server controls and user controls at both design time and run time.

What you’ll learn
You will be able to implement and attach event handlers to web server controls. You will also be able to create a user control and work with its properties and events. After completing this Drill, you’ll also have gained the skills necessary to make web controls more accessible.

Core development tasks
• configure basic web server controls and examine how ASP.NET converts these controls to HTML at run time
• create event handlers and attach them to server control events at design time
• utilize multiple types of list boxes when creating an e-mail contact list
• add web server controls to a Web Form dynamically, and set their properties and event handlers programmatically at run time
• make web server controls more accessible to users with disabilities
• create a user control that can be dragged onto a web page at design time, and set properties and respond to events from that control

What you’ll cover
• Introduction to ASP.NET web server controls
• Using the TextBox, Button and Image controls
• Using web server control events and implementing Page.IsPostBack property
• Working with the Command and SelectedIndexChanged events
• Working with the ListBox, DropDownList and CheckBoxList server controls
• Dynamically creating web server controls
• Adding a Label and a RadioButtonList control to a PlaceHolder control at runtime
• Web control accessibility
• Setting the tab order and access keys on server controls
• User controls and adding web server controls to a user control, which in turns gets added to a web page

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development and server page technology would also be useful.
Drill: State Management and Page navigation

Overview
The ASP.NET Framework provides several options for managing state, which include keeping information on the client and storing information on the server between round trips. This Drill allows you to explore these options in detail.

Audience
This Drill is for developers who want to be able to understand and work comfortably with state management techniques in ASP.NET 2.0.

What you’ll learn
You will be able to implement various state management techniques in applications using HttpCookies, ViewState, and session state. You will have gained the ability to maintain page state after cross-page posting to another page has occurred. In addition, you will be able to implement the caching feature in an ASP.NET application and use it in association with SQL cache dependency.

Core development tasks
- use HttpCookies for state management and use Request to read HTTP POST and GET parameters
- use ViewState for managing persistent user state in your ASP.NET application
- create an ASP.NET web application that maintains access to the page state after cross-page posting to another page
- use the Session object to persist data in ASP.NET
- persist data using the Cache object and to create SQL Server-based dependencies on cached items

What you’ll cover
- State management with HttpCookies and reading parameters from Request
- Using members of the Request and Response objects
- Working with the HttpCookie class
- State management with View State
- Adding to and retrieving from the View State
- Implementing cross-page posting and state management
- Managing session state and using the HttpSessionState class
- Using SQL Cache Dependency and the SQLCacheDependency class
- Working with the caching element of the Web.Config file

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development, server page technology and basic knowledge of SQL Server would also be useful.
Drill: Exception Handling and User Input Validation

Overview
Applications must be able to handle errors that occur during execution uniformly. The tasks in this Drill let you work on useful structured error-handling techniques in your web application.

Audience
This Drill is for developers who want to be able to implement structured exception handling techniques and validate user input in ASP.NET 2.0 applications using validation controls.

What you'll learn
You will be able to implement structured exception handling techniques and validate user input in ASP.NET 2.0 applications using validation controls.

Core development tasks
- implement error handling to throw and catch exceptions of specific types
- create, throw, and catch a custom exception and implement application-level exception handling
- enforce mandatory fields and perform data comparison and range checks using ASP.NET validation controls
- validate the format and content of user input using regular expressions and custom validation controls
- validate parts of a web form using validation groups

What you'll cover
- Structured exception handling and working with the ArgumentException, FileNotFoundException, Exception and Stream classes
- Creating a custom exception and adding the Serializable attribute
- Throwing and catching a custom exception in your application
- Handling ASP.NET application-level errors and working with the customErrors element of the Web.Config file
- Implementing Global.asax and using the HttpServerUtility and Exception classes
- Validating user input in Web Forms and implementing the RequiredFieldValidator, CompareValidator, RangeValidator, and Validation summary controls.
- Using the Page.IsValid property and validating user input using regular expressions and custom validation controls
- Implementing the RegularExpressionValidator, CustomValidator, and Validation summary controls
- Validation Groups and assigning the ValidationGroup property on a number of web server controls
- Implementing the RequiredFieldValidator, RegularExpressionValidator, and Validation summary controls

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development and server page technology would also be useful.
Drill: Rich Controls and Services

Overview
This Drill lets you work with some of the rich controls in ASP.NET 2.0 (including web wizards) and highlights useful techniques such as generating images dynamically.

Audience
This Drill is for developers who want to learn about some of the new rich server controls available in ASP.NET 2.0 and how to use them in their applications. It is also for those who want to learn how to implement localization in an ASP.NET 2.0 application.

What you’ll learn
You will be able to implement many of the rich controls available in ASP.NET 2.0 such as the Wizard, FileUpload, AdRotator, Calendar, TreeView, Object Data Source, and FormView. In addition, you’ll be able to localize a web application by creating resource files. You will also have learned how to use client callbacks to update server controls without causing a full page postback.

Core development tasks
• use the Wizard control to present a data entry process in a structured manner using individual wizard steps and to use the FileUpload control to send files to the server
• use the AdRotator and Calendar Web server controls to create an advertisement web page and to track and gather usage information
• create a TreeView control that is data bound to a hierarchical data source control
• use the ObjectDataSource and FormView web server controls to bind to and display business objects to the user for manipulation
• create localized resources for French language cultures, both for data and UI elements
• use client callbacks to update server controls without causing a full page postback

What you’ll cover
• Using the Wizard and FileUpload controls
• Working with the properties and events of the Wizard control
• Using the properties of the FileUpload control
• Using the AdRotator and Calendar controls
• Using the TreeView control, the DataSource component and data controls
• Working with the ObjectDataSource component and FormView control
• Localization and implementing localization in an application through resource files
• Client callbacks

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development and server page technology would also be useful.
Drill: Master Pages and Site Navigation

Overview
This Drill lets you reuse common code within pages using master pages. Sitemap navigation, sitemap aware controls, and custom provider techniques are also covered.

Audience
This Drill is for developers who want to be able to implement master pages in an ASP.NET 2.0 application and use its navigation web server controls.

What you’ll learn
You will be able to implement master pages that allow you to define an overall look for every page of your application using just one file. You will also be able to easily create web page navigation structures using the new site navigation controls and features available with ASP.NET 2.0.

Core development tasks
• define the overall layout of your web application using a master page
• manipulate the properties and controls of a master page from content pages at runtime
• use the navigation web server controls and built-in SiteMap providers
• filter site-map nodes based on authorization rules
• programmatically modify the site navigation data returned by site-map providers
• implement and use a custom file system site-map provider

What you’ll cover
• Using master pages and working with .master files
• Associating a web form with a master page
• Using the Content and ContentPlaceHolder controls
• Working with master page properties and controls
• Using the Page_PreInit event handler of the Page class
• Using the CustomMaster and LayoutMaster classes
• Using the navigation web server controls
• Working with the Menu, Treeview, SiteMapPath, and SiteMapDataSource controls
• Using .sitemap files and working with the sitemap node of the Web.Config file
• Site navigation security and using the site navigation API
• Working with the SiteMap and SiteMapNode classes
• Creating a custom site map provider

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development and server page technology would also be useful.
Drill: Personalization and Themes

Overview
This Drill enables you to work on the personalization features of ASP.NET 2.0. Using realistic programming scenarios, it lets you apply and personalize themes to Web applications and allow users to choose and create color schemes.

Audience
This Drill is for developers who want to be able to understand and work comfortably with profiles and themes in ASP.NET 2.0 to enable users to personalize the application.

What you’ll learn
You will be able to create and apply themes to an entire ASP.NET 2.0 application or individual pages or controls. You’ll also be able to define and store unique user information using the new Profile feature of ASP.NET 2.0.

Core development tasks
- define and apply a theme to all pages in a web application
- create a themeable user control and apply a skin to it
- define and store per-user settings for use application-wide in an anonymous profile
- create a custom profile base class and configure a web application to use it
- migrate custom profile settings from an anonymous user to an authenticated account
- override the default autosave behavior for profiles using complex property types

What you’ll cover
- Introduction to themes and working with .skin files
- Working with the pages element of the Web.Config file
- Using themes and skins in user controls
- Storing user profiles
- Using the ASP.NET SQL Server Registration Tool to create a database to store profile data
- Creating a custom profile base class and inheriting from the ProfileBase class
- Applying the SettingsAllowAnonymous and SettingsSerializeAs attributes to a property
- Migrating anonymous profiles
- Implementing the Profile_OnMigrateAnonymous event handler in the global application class
- Profile autosaving and implementing the Profile_ProfileAutoSaving event handler in the global application class

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development and server page technology would also be useful.
Drill: Web Parts

Overview
ASP.NET Web Parts are an integrated set of controls for creating web sites, which enable end users to modify the appearance and behavior of web pages directly in a browser. This Drill helps you create Web Parts and, using specific controls, enable end users to add and modify controls at run time. Tasks also cover communication, security, and personalization aspects of Web Parts.

Audience
This Drill is for developers who want to be able to implement Web Parts in their ASP.NET 2.0 applications so they can modify the appearance and behavior of their application in the browser.

What you’ll learn
You will be able to implement Web Part controls for pages in various display modes such as browse mode, design mode, catalog mode, and edit mode. You’ll also be able to show and manipulate Web Parts based on user roles and restore a page to its original state.

Core development tasks
- create and use Web Parts on a Web Form, using both browse and design modes
- use the CatalogZone control to add Web Parts controls to the page at run time
- employ the EditorZone Web Part to modify Web Parts controls at run time
- implement a static Web Parts connection for communication between Web Parts controls
- use the security features of ASP.NET 2.0 to filter Web Parts on a web page
- personalize Web Parts in both shared mode and user mode

What you’ll cover
- Working with the WebPartManager and WebPartZone controls
- Adding the Login and Calendar controls to a WebPartZone control
- Using the CatalogZone Web Part and the CatalogZone and DeclarativeCatalogPart controls
- Adding user controls to a CatalogZone control
- Using the EditorZone Web Part and the EditorZone, AppearanceEditorPart, and PropertyGridEditorPart controls
- Applying the Personalizable and WebBrowsable attributes to a property
- Communication between Web Parts and defining a static Web Part connection between two Web Part controls
- Creating a provider and a consumer connection point using the ConnectionProvider and ConnectionConsumer attributes
- Web Parts security and filtering, and using the AuthorizationFilter property on Web Part controls
- Implementing an event handler for the AuthorizeWebPart event
- Personalization in Web Parts, working with shared and user scope, and resetting the state of a page containing Web Parts

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development and server page technology would also be useful.
Drill: Creating ASP.NET 2.0 Controls

Overview
The tasks in this Drill introduce ASP.NET user controls, which enable you to define your own controls and supplement ASP.NET’s server controls.

Audience
This Drill is for developers who want to be able to create their own controls and use them in their ASP.NET 2.0 applications. These custom controls can be comprised of existing server controls, derived from an existing server control, or created from scratch. It is also for developers who would like to know how to add designer support to their custom control or be able to store its state.

What you’ll learn
You will be able to create a user control encapsulating existing server controls, create and customize a control derived from an existing web control, and create a control derived from the CompositeControl class. You’ll also be able to persist your custom control’s information using the new ControlState property and implement designer support in custom controls.

Core development tasks
- encapsulate existing user interface functionality into a user control so it can be reused within the application
- create a server control that derives from an existing server control to add custom functionality
- generate a server control that derives from the CompositeControl class and is composed of several built-in controls
- build a custom server control that uses control state to persist critical state data across postbacks
- add support for design-time features such as smart tags and auto-formatting to a custom server control
- create a custom server control that uses templates to allow design-time customization

What you’ll cover
- Creating a user control comprised of existing web server controls
- Including an ASP.NET user control in your web page
- Deriving from an existing web control
- Adding custom functionality to a server control derived from the TextBox server control
- Composite controls
- Inheriting the CompositeControl class and overriding its RecreateChildControls method
- Overriding the CreateChildControls and RenderChildren methods of the Control class
- Control state
- Using the ControlState property to persist information specific to your control
- Overriding the OnInit, LoadControlState, and SaveControlState methods of the Control class
- Adding designer support
- Adding the design time smart tags and autoformatting features to a custom server control
- Working with the DesignerAutoFormatCollection and DesignerActionPropertyItem classes
- Applying the Designer attribute to a composite control
- Templated controls
- Applying the PersistenceMode and TemplateContainer attributes to properties of a custom control

**Prerequisites**
We assume that you’re already familiar with the concepts of programming and you've some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development and server page technology would also be useful.
Drill: Reading and Writing XML Data

Overview
This Drill demonstrates how to work with XML data in ASP.NET 2.0. You cover reading and writing XML data, modifying XML files, and validating an XML document against a schema definition.

Audience
This Drill is for developers who want to build ASP.NET 2.0 applications that interact efficiently with data stored in XML format, a versatile and widely used means of storing information.

What you'll learn
You will learn how to work effectively with XML data in ASP.NET 2.0. The skills you will learn include reading and writing XML data, modifying XML files using XmlDocument, and validating an XML document against a schema definition document. You will also learn how to reference external functionality from an Extensible Stylesheet Language Transformations (XSLT) style sheet, and query and format XML data using XML Path Language (XPath) and XSLT transformation.

Core development tasks
- use the XmlReader with XmlReaderSettings and XmlWriter with XmlWriterSettings classes to read and write XML data
- enable an XmlReader to validate an XML document against a schema definition document using appropriate XmlReader settings
- modify the contents of an XML document using the XmlDocument class and the Document Object Model (DOM)
- use the XPath objects exposed by .NET Framework to query an XML file
- develop an understanding of XSLT using C# and XSLT definition language

What you'll cover
- Reading and writing XML data using the XMLReader and XMLWriter objects
- Using the new XmlReaderSettings and XmlWriterSettings classes
- Validating XML files using an XML schema definition document (XSD)
- Configuring an XmlReader to validate against an XSD and modifying XML files using XmlDocument and the DOM
- Querying an XML data file using XPath and the .NET Framework
- Extracting data from an XML document using the XPathNavigator and XPathDocument objects with XPath queries
- Transforming XML documents using XSLT stylesheets with components provided by the .NET Framework
- Converting data from an XML document into HTML using the new XmlCompiledTransform
- Using XSLT extension objects to implement custom routines that process data and perform calculations

Prerequisites
We assume that you’re familiar with the concepts of programming and you’ve some experience with Visual Studio. You’ll also need to be comfortable with Visual C# or Visual Basic syntax and have a working knowledge of XML syntax. A basic understanding of XSLT and XPath is also required.
Drill: Consuming and Manipulating data with ADO.NET

Overview
This Drill demonstrates how to manage data using ADO.NET. You will complete challenges in querying a SQL Server database, manipulating a DataSet, and working with disconnected DataSets. You will also create a strongly typed DataSet, use the DataSet's built-in XML features, and produce code that works with multiple data providers.

Audience
This Drill is for developers who want to use ADO.NET technology to access data sources such as SQL databases, particularly those who want to gain experience using DataSets.

What you'll learn
You will learn the skills needed to manage data using ADO.NET. The skills you will learn include querying a SQL Server database, manipulating a DataSet, and working with disconnected DataSets.

Core development tasks
- execute commands against a SQL Server application via inline SQL and stored procedures using the classes of the System.Data.SqlClient namespace
- populate a DataSet with data from a SQL Server database and modify the DataSet offline before persisting the changes back to the database
- allow data to be modified in a disconnected manner and efficiently written back to the database where updates are performed by a separate, remote component
- create a strongly typed DataSet, and save, load, and synchronize XML data with DataSets
- use the generic classes provided by the System.Data.Common namespace to produce code that works with multiple data providers

What you'll cover
- Querying a SQL Server Database from a .NET application
- Using classes of the System.Data.SqlClient namespace to execute commands against a SQL Server database
- Creating and manipulating a DataSet, and working with classes in the System.Data and System.Data.SqlClient namespaces
- Working with disconnected DataSets and sending modified rows to a web service method using the GetChanges method of the DataSet class
- Creating strongly typed DataSets using the Dataset Designer and XML
- Maintaining a local copy of database data in XML format using the XML features of the DataSet and DataTable classes
- Writing provider-independent code in ADO.NET using the System.Data.Common namespace

Prerequisites
We assume that you're familiar with the concepts of programming and you've experience with Visual Studio. You'll also need to be comfortable with Visual C# or Visual Basic syntax and have a good knowledge and understanding of database concepts with SQL Server. You'll need to know Windows Forms, and previous experience of ADO.NET 1.0 is useful for appreciating the new features of ADO.NET 2.0.
Drill: Data Source Controls and Data Binding

Overview
This Drill shows how to use ASP.NET 2.0's powerful data source controls, which act as intermediaries between an application's interface controls and the underlying data. Tasks show how the SqlDataSource control interacts with a SQL database, how the ObjectDataSource control relates to business objects, and how an XmlDataSource control can display hierarchical information from an XML file.

Audience
This Drill is for developers who want to gain experience using ASP.NET 2.0's versatile data source controls, which are used to bind data sources to the controls that display the information.

What you'll learn
You will learn the skills needed to work with ASP.NET 2.0's data source controls, which act as intermediaries between an application's interface controls and the underlying data. You'll learn how to use an SqlDataSource control with a SQL database, how to relate an ObjectDataSource control to business objects, and how to use an XmlDataSource control to display hierarchical information from an XML file.

Core development tasks
- bind database information to a drop-down list using an SqlDataSource control
- auto-generate statements, reduce concurrency conflicts, and cache data using a SqlDataSource control
- use an ObjectDataSource component as a representation of a business object that provides data to data-bound controls
- pass a parameter to the Select method of an ObjectDataSource component and to use its ObjectCreating event to explicitly set the instance of the business object
- display hierarchical information using data binding with nested Repeater controls that are bound to an XmlDataSource control
- configure an ObjectDataSource control to enable paging using a provided business object's method

What you'll cover
- Adding an SqlDataSource control to a page
- Binding database information to a drop-down list using an SqlDataSource control
- Using wizards to configure a data source
- Configuring an SqlDataSource control to insert, update, and delete database information
- Reducing concurrency conflicts
- Caching information loaded into an SqlDataSource control
- Using an ObjectDataSource component to represent a business object
- Displaying information with a DetailsView control
- Passing a parameter to the Select method of an ObjectDataSource component
- Using the ObjectDataSource's ObjectCreating event to explicitly set the instance of a business object
- Displaying hierarchical information using data binding
- Using nested Repeater controls bound to an XmlDataSource control
- Binding an XmlDataSource control to an XML file
- Configuring an ObjectDataSource control to enable paging
• Implementing paging in the business layer to reduce database traffic

**Prerequisites**
We assume that you’re familiar with the concepts of programming and you’ve experience with Visual Studio, including its designer and control wizards. You’ll also need to be comfortable with Visual C# or Visual Basic syntax and have a working knowledge of web server controls. You’ll need to know the syntax of SQL, data-binding, and XML Path Language (XPath). A basic understanding of XML is also required.
Drill: Using Data-bound controls

Overview
This Drill shows how to use ASP.NET 2.0's versatile data-bound controls which enable you to create highly functional user interfaces with relatively little coding. Several tasks demonstrate different aspects of the versatile GridView control. Other tasks show how to implement a master/detail page, and how to create different views of the underlying data using the DetailsView, FormsView, and TreeView controls.

Audience
This Drill is for developers who want to efficiently create rich user interfaces using ASP.NET 2.0's data-bound controls, particularly those who want to gain experience using the GridView control.

What you'll learn
You will learn the skills needed to use ASP.NET 2.0's data-bound controls, which enable the creation of highly functional user interfaces with relatively little coding. You'll learn several techniques for using the versatile GridView control. You will also learn how to implement a master/detail page, and how to create different views of the underlying data using the DetailsView, FormsView, and TreeView controls.

Core development tasks
- use a GridView control with an ObjectDataSource control to enable paging, editing, and deleting of records
- manipulate rows, cancel events, handle exceptions, and configure the HyperLinkField object within a GridView control
- implement sorting functionality using ObjectDataSource and GridView controls
- create a master/detail page using a combination of DropDownList, GridView, and DetailsView controls
- use the BulletedList and FormView web server controls new to .NET Framework 2.0
- bind a TreeView control to an XML file through an XmlDataSource control, and display the information from the XML file hierarchically

What you'll cover
- Adding a GridView control to a page and binding it to an ObjectDataSource control
- Configuring a GridView control to enable paging, editing, and deleting of records
- Configuring a GridView control to prevent the entire page from being posted back to the server
- Accessing and manipulating rows as they are data-bound
- Cancelling events and handling exceptions in the GridView control
- Configuring a HyperLinkField control
- Implementing sorting functionality with the GridView control
- Displaying information about the last sort to the user
- Creating a master/details page
- DropDownList and DetailsView controls
- Adding a BulletedList control to a page
- Combining the BulletedList and FormView controls
- Binding a TreeView control to an XmlDataSource control
- Setting the number of tree levels bound to a TreeView control
- Specifying the events to be raised when a node is selected
Prerequisites
We assume that you’re familiar with the concepts of programming and you’ve experience with Visual Studio, including its designer and control wizards. You’ll also need to be comfortable with Visual C# or Visual Basic syntax and have a working knowledge of web server controls. You’ll need to understand exception handling and data-binding syntax. A basic understanding of XML is also required.
Drill: Membership, Login Controls and Forms Authentication

Overview
Using real-world programming scenarios, this Drill lets you work with the features of the Membership provider. You will work with Login Controls (a new control available in ASP.NET 2.0) and discover what’s new in Visual Studio Forms Authentication.

Audience
This Drill is for developers who want to be able to understand and work with the Membership provider in ASP.NET 2.0. It is also for those who want to work with Login server controls and forms authentication in ASP.NET 2.0.

What you’ll learn
After taking this Drill, you will be able to implement the various Login controls available in ASP.NET 2.0 and understand when to use each one. You will be able to create, store, update, and delete membership information for users of a web site either through the Login controls, the Web Site Administration Tool, or through the members of the Membership class. You will also be able to create your own custom provider if the default providers do not support your site’s data source.

Core development tasks
• use the ASP.NET 2.0 configuration tools and Web.config file to configure the membership provider
• authenticate a web site using both simple and advanced login controls
• use the membership API to programmatically manipulate user information
• create a membership provider for a custom data source

What you’ll cover
• SQL Server Membership provider
• ASP.NET SQL Server Registration Tool
• ASP.NET Web Site Administration Tool
• Working with Login server controls in ASP.NET 2.0
• The Membership API and working with the various members of the Membership class
• Adding, updating, deleting, and retrieving information about users from the membership database
• Creating a custom MembershipProvider and using the members of the MembershipProvider class

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development, server page technology and basic knowledge of SQL Server would also be useful.
Drill: Authorization and Roles

Overview
Using real-world programming scenarios, this Drill lets you work with users and roles in ASP.NET 2.0, including their authorization and management. At the end of the Drill, you will also perform techniques involving role-based code accessed security.

Audience
This Drill is for developers who want to be able to understand and work with users and roles in their ASP.NET 2.0 applications.

What you’ll learn
After taking this Drill, you will be able to create and assign users and roles using either the ASP.NET Web Site Administration tool or using the Role class. Using URL Authorization, you will be able to permit or deny access to specific parts of the web application. You will be able to create a custom role provider when your data source does not satisfy the default providers and you will be able to restrict permission to certain parts of the source code in an application based on roles.

Core development tasks
• use the role manager give role grouping to the web site security
• use the Membership and Roles classes to maintain users and roles
• discover how configuration settings influence URL authorization in ASP.NET applications
• implement and use a custom role provider
• control user access to application functionality by defining permission requirements

What you’ll cover
• Forms Authentication and Role Management
• ASP.NET Web Site Administration Tool
• Working with Login server controls in ASP.NET 2.0
• The Membership API and working with the members of the Membership class
• Working with the members of the Roles Class
• URL Authorization
• Restricting access to web resources using the authorization element in the Web.Config file
• Creating a custom role provider and using the members of the RoleProvider abstract class
• Role-Based code access security
• Using PrincipalPermissionAttribute attributes to restrict permission and using the members of the PrincipalPermission class

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development, server page technology and basic knowledge of SQL Server would also be useful.
Drill: Web Application Attacks

Overview
Using real-world programming scenarios, this Drill helps you develop techniques for defending your web application against attacks from malicious users. You will prevent attacks caused by malicious input on web forms that can cause SQL injection or cross-site scripting attacks. You will also implement measures to protect against unauthorized access of resources in your application.

Audience
This Drill is for developers who want to learn about and defend against some of the common attacks that web applications face from malicious users.

What you'll learn
After taking this Drill, you will be able to implement various techniques in web applications to correctly validate any input received via web forms, and prevent against SQL injection, cross-site scripting, canonicalization, denial of service, and view state attacks. You will also understand why preventing these types of attacks is so important.

Core development tasks
- detect and counter application attacks by using input validation
- defend an application against SQL injection attacks
- protect HTML markup enabled input fields against cross-site script attacks
- use canonicalization to secure resources in a system where access is based on file names
- counter denial of service attacks at the application level and defend against view state attacks

What you'll cover
- Validating input on web forms and using Validator server controls
- Using regular expressions to constrain and validate input
- Preventing SQL injection attacks
- Using Parameterized SQL statements
- Configuring a custom error page using the customErrors element in the Web.config file
- Preventing cross-site scripting attacks
- Encoding content and protecting cookies from client-side access by creating HTTP-only cookies
- Canonicalization & one-click attacks
- Using the List Generic and Uri classes and the MembershipUser.LastLockoutDate property
- Using the MembershipProvider.UnlockUser method
- Restricting file upload size in a web application using the httpRuntime element in the Web.Config file
- Preventing view state attacks and encrypting view state using Advanced Encryption Standard (AES)
- Storing Session ID in the view state

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development, server page technology and basic knowledge of SQL Server and the T-SQL language would also be useful.
Drill: Securing and Monitoring ASP.NET Applications

Overview
Using real-world programming scenarios, this Drill allows you to work with various areas of authorization and authentication in the application. You will deal with partially trusted code, monitor the health of your ASP.NET application, and implement security events.

Audience
This Drill is for developers who want to be able to use techniques to secure vulnerable sections and monitor the health of their ASP.NET 2.0 applications.

What you’ll learn
After taking this Drill, you will be able to encrypt vulnerable sections of an application where data such as database connection strings are stored using members of the DPAPI. You will also be able to restrict access to sensitive file and folders using a combination of Windows authentication and URL authorization. You will have the skills to enable a partially-trusted application to access secure operations. You will also be able to extend the security of your application by adding pre- and post-processing to incoming HTTP requests, and track security-related web events using health monitoring and instrumentation features.

Core development tasks
- use the Windows Data Protection API (DPAPI) and the Aspnet_regiis.exe tool to encrypt sensitive data in your configuration files
- access a fully trusted assembly from a partially trusted web application.
- use Windows authentication and URL authorization to restrict access to certain files and folders
- create a custom HttpModule to authenticate requests to a web application against a custom authentication service
- carry out health monitoring and instrumentation of Security events

What you’ll cover
- Using the Windows Data Protection API (DPAPI)
- ASP.NET IIS Registration command line tool
- Encrypting the connectionStrings section of a Web.Config file
- Using members of the ProtectedData class
- Storing values in session variables
- Running an application in medium trust mode
- FileIOPermission attribute
- Signing an assembly with a strong name
- Adding an assembly to the Global Assembly Cache
- Windows Authentication
- URL Authorization
- Using members of the WindowsIdentity class
- Using the PrincipalPermissionAttribute attribute
- Allowing and denying access to users and roles in the Web.Config file
- Creating a custom authentication mechanism
- Implementing the IHttpModule interface
- Using the GenericPrincipal and GenericIdentity classes
- Adding a custom authentication service to the HTTP pipeline through the httpModules element in the Web.Config file

Drill Details

Status
Released

Developer Program
ASP.NET Web Development with Visual Studio

Duration
2.5 hours

Programming Language
C# & VB

System Requirements
- A non-express edition of Visual Studio
- Administrator rights for the development machine
- SQL Server

Level
3

Drill Code
- D0019D-CS
- D0019D-VB
- Configuring predefined health monitoring events to instrument your application to track security-related events
- Creating and raising a custom event to monitor additional security-related activity
- Configuring the healthMonitoring section of the Web.Config file

**Prerequisites**
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development, server page technology, and basic knowledge of SQL Server would also be useful.
Drill: Styling Websites with Cascading Stylesheets (CSS)

Overview
Cascading Style Sheets (CSS) is a simple mechanism for adding fonts, colors, spacing, and other style elements to Web documents. This Drill covers a number of important techniques required to style Websites effectively with CSS.

Audience
This Drill is for developers who want to use Cascading Style Sheets (CSS) techniques to add fonts, colors, spacing, and other style elements to Web documents.

What you’ll learn
After taking this Drill, you’ll be able to use CSS to create both flexible boxes with rounded corners and an application menu. You’ll also apply flexible fonts to a web document and refactor an HTML table layout to a three column layout. What’s more, you’ll learn how to style both the ASP.NET GridView control and a web form.

Core development tasks
- apply flexible fonts to a web document
- refactor an HTML table layout
- create boxes with rounded corners
- style an application menu and enforce browsers to use the standard box model for rendering elements
- style a HTML table generated by the ASP.NET GridView control
- style a web form that uses fieldset and definition-list HTML elements

What you’ll cover
- Using basic CSS selectors and style rules to apply flexible, scalable fonts to a web document
- Separating the style information in a style sheet document
- Linking a style sheet to a web document
- Using div elements to refactor an HTML page
- Using the float, width, and padding CSS properties
- Using CSS with HTML markup to create boxes with rounded corners
- Applying more than one CSS class to an element
- Dealing with rule conflicts
- Creating a horizontal navigation menu by using CSS
- Instructing ASP.NET to generate XHTML Strict markup
- Informing browsers about DOCTYPE (document type definition) and the rendering mode for generated pages
- Styling a HTML table generated by an ASP.NET GridView control
- Styling individual columns in a table
- Suppressing line breaks inside a table cell
- Vertically and horizontally aligning the content of a table column
- Using CSS to style an HTML form created using the fieldset and definition list elements
- Working with properties used for positioning and layout of elements
- Changing the display type of the mouse cursor
Prerequisites
We assume that you're already familiar with the concepts of programming and you've some experience with Visual Studio. For this particular Drill, you'll also need to be comfortable with ASP.NET applications, HTML markup, and CSS syntax.
Developing Web Applications with AJAX using Visual Studio

This Developer Program contains the following Drill:

- The Microsoft ASP.NET AJAX Extensions

The following books cover the recommended prerequisites for this particular Developer Program:

- *Programming ASP.NET AJAX* by Christian Wenz, published by O'Reilly Media, ISBN 0596514247
Drill: The Microsoft ASP.NET AJAX Extensions

Overview
The tasks in this Drill demonstrate how to enhance a website with Microsoft’s ASP.NET AJAX technology. It covers important AJAX controls such as UpdatePanel, ScriptManager and ScriptManagerProxy, Timer, and UpdateProgress, as well as sending AJAX requests to web services.

Audience
This Drill is for developers who want to learn how to use Microsoft's ASP.NET AJAX technology to enhance website responsiveness and usability.

What you'll learn
You will learn how to use Microsoft’s AJAX technology to improve website performance and enhance the user experience. You will be able to use the important ScriptManager and UpdatePanel controls to AJAX-enable a website. You’ll also find out how to refresh a control's content, provide progress feedback to the user, and send an AJAX request to a web service.

Core development tasks
- AJAX-enable an ASP.NET website
- control page updates with the UpdatePanel control
- include custom JavaScript scripts with the ScriptManagerProxy
- use the UpdateProgress control
- use the Timer control to send AJAX requests to web services

What you'll cover
- Using ScriptManager & UpdatePanel controls and enabling AJAX-style partial page updates
- Using UpdatePanel to control when page updates occur
- Using ScriptManager and ScriptManagerProxy controls with Master/Content pages
- Referencing custom scripts with ScriptsReference & adding an UpdateProgress control to a Web Form
- Refreshing a control's content using a Timer control
- Sending AJAX requests to a web service
- Generating and using JavaScript proxies for .NET server-side types

Prerequisites
We assume that you’re already familiar with the concepts of programming and you've some experience with Visual Studio. For this Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax, and ASP.NET. Although not essential, some knowledge of JavaScript would also be useful.
Enterprise Library for .NET Framework 2.0 using Visual Studio

This Developer Program contains the following Drills:

- Working with the Enterprise Library Application Blocks Part 1
- Working with the Enterprise Library Application Blocks Part 2
Drill: Working with the Enterprise Library Application Blocks Part 1

Overview
This Drill will develop your skills in using several Enterprise Library application blocks. Tasks include configuring and using the Data Access Application Block, tracing activities with the Logging Application Block, and using the Exception Handling Application Block to catch and throw exceptions.

Audience
This Drill is for developers who want to learn how to use the Data Access, Logging, and Exception Handling application blocks of the Enterprise Library.

What you'll learn
You will learn how to use several Enterprise Library application blocks to address common development challenges. The skills you will learn include using and configuring the Data Access Application Block to access a database. You will also find out how to trace activities with the Logging Application Block, and how to throw, catch, and log exceptions with the Exception Handling Application Block. After taking this Drill, you will also be able to use the Enterprise Library Configuration Console.

Core development tasks
• use the Data Access Application Block to retrieve and modify data from a database
• employ the Logging Application Block to log information to a file and trace activities to a database
• configure the Exception Handling Application Block to catch, wrap, and throw exceptions
• configure the Exception Handling Application Block and the Logging Application Block to catch and log exceptions

What you'll cover
• Configuring application blocks with the Enterprise Library Configuration Console
• Using the Data Access Application Block to retrieve information from a database
• Using the Database and ADO.NET DbCommand classes
• Logging information to a file with the Logging Application Block
• Configuring a flat file trace listener
• Saving log information to a variety of destinations and tracing user interactions
• Using the Exception Handling Application Block to handle exceptions such as InvalidOperationException, ArgumentNullException, and ComException
• Defining an exception handling policy, wrapping and rethrowing exceptions and logging exceptions in different formats

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio. For this Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax, XML syntax, SQL syntax, as well as SQL databases (Express or Server). Although not essential, experience with individual application blocks or previous releases of Enterprise Library would also be useful.
Drill: Working with the Enterprise Library Application Blocks Part 2

Overview
This Drill will develop your skills in using several Enterprise Library application blocks. Tasks include caching data with the Caching Application Block; using the Cryptography Application Block to encrypt, decrypt, and hash data; and configuring authorization and caching user identities with the Security Application Block.

Audience
This Drill is for developers who want to use application blocks from the Enterprise Library to cache, encrypt, and secure application data during enterprise development.

What you’ll learn
You will learn the skills required to work with the Caching Application Block, the Cryptography Application Block, and the Security Application Block. You will find out how to cache frequently accessed data, as well as encrypt, decrypt, and hash data. You will also learn how to configure user authorization and cache user identities.

Core development tasks
• use the Caching Application Block to cache frequently accessed data
• configure the Cryptography Application Block to encrypt and decrypt data using the same cryptographic key
• configure and use the Cryptography Application Block to hash data using a salt, and compare plain text values with their hash equivalent
• employ the security application block for authorization
• set the security cache manager to cache user identities

What you’ll cover
• Configuring the Caching Application Block to store frequently accessed database information in the in-memory cache
• Configuring the Caching Application Block to cache data to isolated storage
• Encrypting and Decrypting data using the System.Security.Cryptography.RijndaelManaged algorithm type and a cryptographic key
• Configuring the App.config file to use the hashing functionality of the Cryptography Application Block
• Working with the CreateHash and CompareHash methods of the Cryptographer facade
• Configuring an application to use an authorization rule provider from the Security Application Block
• Using the Security Application Block to configure an instance of a security cache provider that can store user identities

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio. For this Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax, XML syntax, SQL syntax, as well as SQL databases (Express or Server). Although not essential, experience with individual application blocks or previous releases of Enterprise Library would also be useful.
Object-Oriented Design & Development with Visual Studio

This Developer Program contains the following Drills:

- Object-Oriented Development Fundamentals
- Input-output
- Generics/Collections
- System Services
- The .NET Runtime
- Events and Structural Design Patterns
- Behavioral Design Patterns
- Creational Design Patterns
- Enterprise Patterns
- Domain-Driven Design Patterns

The following books cover the recommended prerequisites for this particular Developer Program:

- **Design Patterns in C#** by Steven John Metsker, published by Addison-Wesley Professional, ISBN 0321126971
- **Head First Design Patterns** by Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra, published by O'Reilly Media, ISBN 0596007124
- **Patterns of Enterprise Application Architecture** by Martin Fowler; David Rice; Matthew Foemmel; Edward Hieatt; Robert Mee; Randy Stafford, published by Addison-Wesley Professional, ISBN: 0321127420
Drill: Object-Oriented Development Fundamentals

Overview
This Drill introduces some of the key concepts of object-oriented development including encapsulation, inheritance, and polymorphism, all of which enable you to create modular, loosely coupled, and highly reusable software modules.

Audience
This Drill is for developers who want to be able to understand and work with the key concepts of object-oriented development.

What you'll learn
You will learn how to implement important techniques of object-oriented development to create better code. You will be able to improve the structure of your code, enhance class inheritance, and refactor error codes into exceptions. You will also learn how to use interfaces to achieve polymorphism in your applications and specify common behaviors and characteristics for derived classes.

Core development tasks

• enhance the structure of a project by extracting business logic from a form
• implement object-oriented interface polymorphism
• create an abstract class to provide implementations for common functionality
• use the Liskov Substitution Principle to improve class inheritance
• replace inheritance with delegation
• replace an error code with an exception

What you'll cover

• Creating a new class by extracting logic from a form
• Defining an interface
• Creating an abstract class
• Implementing the Liskov Substitution Principle
• Replacing inheritance with delegation
• Replacing an error code with an exception

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax. Although not essential, experience in object-oriented development and familiarity with Generics would also be useful.
Drill: Input-output

Overview
The .NET Framework provides comprehensive support for managing the file system. In this Drill, you use these facilities to handle files and directories, and to extract information using regular expressions. The Drill also lets you implement practical techniques for communicating over the network.

Audience
This Drill is for developers who want to be able to work with files and directories, retrieving information and working with the data in the files themselves. It is also for those who want to use practical techniques for communicating over the network via web requests or TCP.

What you'll learn
You will be able to use the classes provided by the .NET Framework for managing the file system to handle files and directories, and extract information using regular expressions. You will also be able to use practical techniques for communicating over the network via web requests to TCP.

Core development tasks
- use the FileInfo and Directory classes to obtain information about individual files
- use File, FileStream, StreamReader, StreamWriter, and DeflateStream to read from and write to text and compressed files
- use regular expressions to retrieve specific information from a text file
- use regular expressions to extract specific information from an HTML file
- use the WebRequest and WebResponse classes to make simple requests to web sites
- connect to a remote TCP host, to send and receive data from it, and to retrieve information about a specific host from the Internet Domain Name System (DNS)

What you'll cover
- Retrieving file and directory information
- The Directory.GetFiles Method
- Using the members of the FileInfo class
- Using the members of the DirectoryInfo class
- The Path.GetExtension Method
- Reading, writing and compressing files
- Using Statement
- Using the File, FileStream, StreamReader, StreamWriter, and DeflateStream classes
- Retrieving data using regular expressions
- Using the Regex class
- Using the StringCollection class
- Making web requests
- Using the WebRequest and WebResponse classes
- Connecting to Remote TCP host
- Sending and Receiving data
- Retrieving Information about a host from the Internet Domain Name System (DNS)
- The Dns.GetHostEntry Method
- The TcpClient Class

Drill Details

Status
Released

Developer Program
Object-Oriented Design & Development with Visual Studio

Duration
3 hours

Programming Language
C# & VB

System Requirements
C# & VB

Level
2

Drill Code
- D0020B-CS
- D0020B-VB
• The NetworkStream Class
• Async Callback Delegate

Prerequisites
We assume that you're already familiar with the concepts of programming and you've some experience with Visual Studio. For this particular Drill, you'll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development would also be useful.
Drill: Generics/Collections

Overview
Using real-world programming scenarios, this Drill introduces the System.Generics namespace. You use this namespace to create strongly typed collections, stacks, queues, and dictionaries. You also use it to create iterators and your own generic types.

Audience
This Drill is for developers who want to learn about and work with the classes in the System.Generics namespace (new to .NET 2.0). With these classes, you will be able to create strongly typed collections, stacks, queues, and dictionaries.

What you’ll learn
You will be able to create strongly typed collections, stacks, queues, and dictionaries using the classes of the System.Generics namespace. You will be able to use it to create iterators and your own generic types. You will also have the ability to constrain generic types to enforce the requirements that a type must meet before being passed to a generic class.

Core development tasks
- use generics to create a strongly typed list
- add functionality to a generic collection by deriving from it
- create and use iterators to support foreach iteration without implementing the IEnumerable interface
- use a generic dictionary object to create a strongly typed collection of key/value pairs
- use generics to create strongly typed stacks and queues
- place constraints on the type parameters of a generic type

What you’ll cover
- Using a generic list instead of an ArrayList
- The List generic class
- Creating a strongly typed collection using generics
- Using methods of the List generic class
- Iterators
- The IEnumerable Generic interface
- How to use the yield return statement
- Using a generic dictionary and the members of the Dictionary generic class
- Creating strongly typed Stack and Queues and using the members of the Stack and Queue generic classes
- Constraining generic types and using the where and new constraints

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development would also be useful.
Drill: System Services

Overview
You can use .NET Framework system services to perform diagnostics and monitoring in your applications. This Drill allows you to practice real-world techniques on launching external processes, tracing, event log handling, performance counters manipulation, and code access security.

Audience
This Drill is for developers who want to be able to interact with some of the services on their computers such as event logs and the performance monitoring tool from within their .NET applications. It is also for those who want to implement code access security in their windows application and deploy it as a ClickOnce application.

What you’ll learn
You will be able to perform diagnostics and monitoring on your applications by working with external processes, custom event logs, custom trace listeners, and performance counters. You will also be able to implement code access security and deploy a Windows application as a ClickOnce application.

Core development tasks
• launch an external process, examine its properties, and terminate it
• create a custom event log programmatically, write to it, remove the log entries, and delete the event log
• create a custom trace listener to write to a log file, and configure a trace switch to specify the type of message to be written out
• retrieve and display performance counter information programmatically, create a category of custom performance counters, and track application performance
• deploy a ClickOnce application and request appropriate permissions
• restrict permissions for certain methods in your application to reduce the risk of them being misused by a malicious application

What you’ll cover
• Working with external processes
• Using the members of the Process and ProcessStartInfo classes
• Working with custom event logs
• Using the members of the EventLog class
• Implementing a custom trace listener
• Working with the trace element of the App.config file
• Inheriting the TextWriterTraceListener class
• Using the TraceLevel enumeration
• Using performance counters
• Using the members of the PerformanceCounter and the PerformanceCounterCategory classes
• Code access security and ClickOnce deployment
• Applying the DnsPermission and FileIOPermission attributes
• Specifying an applications ClickOnce security settings
• Publishing a ClickOnce application
• Restricting permissions with code access security
• Using the PermissionSet and FileIOPermission classes
Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development would also be useful.
Drill: The .NET Runtime

Overview
This Drill introduces Microsoft .NET runtime programming with real-world scenarios. In this Drill, you complete challenges in memory management, particularly when interoperating with unmanaged code, and dynamically extend components through reflection and configuration files.

Audience
This Drill is for developers, who want to work with useful aspects of the .NET runtime such as reflection, delay signing assemblies, disposing of resources, and platform invocation services.

What you'll learn
You will be able to create a custom configuration section handler in your application as well as dispose of managed and unmanaged resources and provide application extensibility through reflection. You will also have the ability to delay sign an assembly and call external native windows API methods using Platform Invocation.

Core development tasks
- create a custom configuration section handler using the ConfigurationSection, ConfigurationElementCollection, and ConfigurationElement classes
- use delayed signing to sign an assembly with a strong name
- implement Dispose methods and the abstract SafeHandle class to clean up managed and unmanaged resources
- create and use custom attributes with reflection
- use reflection to provide application extensibility
- use Platform Invocation Services (PInvoke) to call external native windows API methods and to employ Component Object Model Interoperation (COM Interop)

What you’ll cover
- Creating a custom configuration section handler
- Working with the configSections element of the App.config file
- Using the ConfigurationProperty attribute
- Inheriting from the ConfigurationElementCollection class
- Delay signing an assembly
- The Strong Name tool
- Extracting a public key from a key-pair
- Disposing of managed and unmanaged resources
- Implementing the IDisposable interface
- Using the abstract SafeHandle class
- Implementing the Dispose pattern
- Calling the GC.SuppressFinalize method
- Creating and using custom attribute with reflection
- Inheriting the Attribute class
- Using the AttributeUsage attribute
- Using the MemberInfo.GetCustomAttributes method
- Providing application extensibility using reflection
- Using the members of the Assembly, Type and Activator classes
• Using Platform Invocation Services and COM component integration
• Adding and using the Preview ActiveX control
• Applying the DllImport attribute

Prerequisites
We assume that you're already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and HTML. Although not essential, experience in object-oriented development would also be useful.
Drill: Events and Structural Design Patterns

Overview
Easing application design by identifying a simple way to realize relationships between entities, structural design patterns describe how to combine classes and objects to form larger structures. This Drill lets you explore topics such as the Adapter, the Decorator, and the Composite design patterns.

Audience
This Drill is for developers who want to be able to understand more about events and the common structural design patterns.

What you’ll learn
You will learn various ways to handle events, and how to apply Structural design patterns such as Adapter, Decorator, Composite, and so on. You will see how these design patterns can be used to dynamically create larger structures from a number of smaller classes working together. Using these patterns, you will be able to create structures that, while complex, are easy to understand and modify.

Core development tasks
- implement a cancelable event and prevent subsequent events from firing using a class derived from the CancelEventArgs class
- perform event handling by overriding a virtual method of a base class
- implement the Move Accumulation to Collecting Parameter design pattern
- employ the Adapter design pattern to convert the interface of a class into another interface that clients expect
- implement the Decorator design pattern to extend the functionality of an object dynamically without modifying the original class or creating a subclass
- use the Composite design pattern to implement a recursive, single-interface class structure

What you’ll cover
- Canceling an event
- Handling events by overriding a virtual method
- Moving accumulation to a collecting parameter
- Implementing the Adapter design pattern
- Using the Decorator design pattern
- Implementing the Composite design pattern

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax. Although not essential, experience in object-oriented development and familiarity with Generics would also be useful.
Drill: Behavioral Design Patterns

Overview
Behavioral design patterns identify common communication patterns between objects and concentrate on the assignment of responsibilities between them. Focusing on the way objects interact rather than the flow of control, this Drill lets you work on behavioral design patterns such as the Template Method, the Command, and the State design patterns.

Audience
This Drill is for developers who want to be able to understand more about events and the common structural design patterns.

What you’ll learn
You will learn how to apply important Behavioral design patterns such as Visitor, Template Method, Strategy, Command, and Chain of Responsibility. You will see how these patterns are based on common behaviors that have been identified between objects. Using these patterns, you will be able to reduce coupling between senders and receivers, reduce code complexity and duplication, and make programs more extensible and easier to support.

Core development tasks
• create a chain of objects with distributed responsibility, without coupling the sender to the receiver
• apply the Visitor design pattern to simplify operations on the elements of an object
• use the Template Method design pattern to create an abstract base class with a template method that includes common functionality, but defers specific behavior to subclasses
• apply the Strategy design pattern to reduce the complexity of a program that uses multiple strategy implementations
• encapsulate a request as an object to reduce coupling between the sender and the receiver
• provide undo and redo functionality using the Command design pattern
• extract state information into a separate class to improve extensibility and avoid duplication

What you’ll cover
• Applying the Template Method design pattern
• Using the Strategy design pattern
• Implementing the Command design pattern supporting undo and redo operations using command objects
• Implementing the State design pattern
• Using the Chain of Responsibility design pattern
• Implementing the Visitor design pattern

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax. Although not essential, experience in object-oriented development and familiarity with Generics would also be useful.
Drill: Creational Design Patterns

Overview
This Drill focuses on Creational design patterns, which deal with object creation mechanisms that avoid design problems such as excessive complexity. Some of the patterns covered include the Singleton, the Builder, and the Abstract Factory design patterns.

Audience
This Drill is for developers who want to learn more about some commonly used creational design patterns which will help avoid design problems such as excessive complexity.

What you'll learn
You will learn how to apply Creational design patterns such as Null Object, Singleton, Builder, Factory Method, and Abstract Factory. You will see how such design patterns can be used to provide the most appropriate object creation mechanisms in different situations. Using these patterns, you will be able to achieve the classic design pattern objective of reducing code complexity, in the context of object creation.

Core development tasks
• use the Null Object design pattern to avoid repeatedly checking for null values within code
• allow the creation of only one instance of a class by implementing the Singleton design pattern
• apply the Builder design pattern to simplify the creation of complex objects
• remove code duplication by encapsulating classes using the Factory Method design pattern
• use the Abstract Factory design pattern to simplify the creation of a group of related objects

What you'll cover
• Implementing the Null Object design pattern
• Creating a Singleton class
• Using the Builder design pattern
• Encapsulating classes using the Factory Method design pattern
• Using the Abstract Factory design pattern

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax. Although not essential, experience in object-oriented development and familiarity with Generics would also be useful.

Drill Details
Status
Released
Developer Program
Object-Oriented Design & Development with Visual Studio
Duration
2.5 hours
Programming Language
C# & VB
Level
3
Drill Code
• D0024D-CS
• D0024D-VB
Drill: Enterprise Patterns

Overview
Enterprise patterns represent architectural building blocks that aid in the design of enterprise applications. This Drill features important Enterprise patterns such as Model View Presenter, Domain Model, Gateway, and Service Locator.

Audience
This Drill is for developers who want to learn more about some commonly used Enterprise patterns, which help to enhance maintainability and avoid excessive complexity in your applications.

What you’ll learn
You will learn how to apply Enterprise patterns such as Model View Presenter, Domain Model, Gateway, and Service Locator. Additionally, you will learn how to create a layered application architecture and use a sevice layer.

Core development tasks
- partition an application's architecture into logically coherent layers
- apply the Model View Presenter pattern to a WinForms application
- use a domain model to model business objects, their interactions and relationships
- consolidate application-specific workflow logic into a well-defined application service
- apply the Gateway pattern to encapsulate and simplify access to a complex API
- apply the Service Locator pattern to transparently locate services in a uniform manner

What you’ll cover
- Partitioning a project into logical layers of presentation, business logic, and infrastructure
- Maintaining loose coupling between the layers of system architecture
- Referencing the business and infrastructure layers from the presentation layer
- Achieving a clean separation between presentation logic and model
- Separating application logic from the UI that drives it
- Expressing an application's business logic as a self-contained software model
- Creating an application service that encapsulates high level application logic
- Implementing the Gateway pattern to prevent clients from dealign with complex APIs
- Providing an intermediary that handles the complexities of the API, while providing a more user-friendly API to clients
- Implementing the Service Locator and Plugin patterns to reduce coupling between software dependencies

Prerequisites
We assume that you’re already familiar with programming concepts and that you’ve used Visual Studio. For this particular Drill, you'll also need to be comfortable with Visual C# syntax and have experience in both object-oriented and web development. Although not essential, some familiarity with design patterns would also be useful.
Drill: Domain-Driven Design Patterns

Overview
Domain-driven Design (DDD) Patterns are reusable solutions to commonly occurring problems. They are used in software models with complex domains. This Drill features important DDD Patterns such as Entities, Value Objects, Repositories, and Factories.

Audience
This Drill is for developers who want to learn more about some commonly-used DDD Patterns, which help to speed up the development process by providing tested, proven development paradigms.

What you’ll learn
You will learn how to apply DDD patterns such as Entities, Value Objects, Repositories, and Factories. Additionally, you will learn how to implement the Specification Pattern, as well as how to cluster objects into Aggregates.

Core development tasks
- distinguish and implement Entities and Value objects, the building blocks of Domain-driven Design
- create an Aggregate from an existing class and hide a related class behind the Aggregate boundary
- use Factories to create a domain object
- create Repositories to mediate between the domain and data mapping layers
- implement the Specification pattern with a repository to control objects returned from a generic repository interface

What you’ll cover
- Distinguishing between Entity and Value objects in a domain
- Making a Value object immutable and adding a unique field to an Entity object
- Implementing the Equals and GetHashCode methods
- Refactoring existing code to create an Aggregate from an existing class, and hiding a related class behind the Aggregate boundary
- Refactoring the constructors of a class into three factory methods to hide the creation rules from the client
- Modifying client code to enable the factories methods to create/re-create document instances
- Defining and implementing the repository interface section of a domain model on an existing repository class
- Modifying the client form to use only the functionality available on the interface
- Creating an implementation of the Specification pattern to interact with an infrastructure layer in a domain specific manner
- Implementing a specific set of requirements via the IsSatisfiedBy method of the specification class
- Using a repository on the infrastructure layer to query a data store for candidate items
- Modifying the repository and the client application to use the Specification pattern in order to simplify both of these layers and improve the domain’s visibility and utility within the system
Prerequisites
We assume that you’re already familiar with programming concepts and that you’ve used Visual Studio. For this particular Drill, you'll also need to be comfortable with Visual C# syntax and have experience in both object-oriented and web development. Although not essential, some familiarity with design patterns would also be useful.
Programming WCF using Visual Studio

This Developer Program contains the following Drills:

- Windows Communication Foundation Fundamentals
- Creating Reliable Services with Windows Communication Foundation
- Windows Communication Foundation Security

The following book covers the recommended prerequisites for this particular Developer Program:

Drill: Windows Communication Foundation Fundamentals

Overview
This Drill introduces you to the fundamental concepts of programming with Windows Communication Foundation (WCF), including defining endpoints in code and in configuration, customizing the DataContract and OperationContracts exposed by a service, and specifying minimum requirements for the binding chosen in configuration.

Audience
This Drill is for developers who want to learn the basic skills needed to use WCF, one of the key components of Microsoft’s new .NET Framework 3.0.

What you’ll learn
You will learn the skills needed to work with WCF, Microsoft’s new unified framework for building secure, reliable distributed applications. This includes different ways to connect a client to a WCF service, such as sharing contracts and using a proxy. You will also learn how to configure a service with multiple endpoints, host services in IIS, define data contracts, and handle service failures.

Core development tasks
- connect to a WCF service using a generated proxy, and call methods on that service
- share contracts with a WCF service
- define a service contract and host a WCF service that implements the contract with multiple endpoints
- host a WCF service in a web project
- define a data contract in a WCF service that enables the service to return a custom type
- specify a SOAP fault that is returned to the client when an operation encounters a problem

What you’ll cover
- Connecting to a WCF service using a generated proxy and calling methods on a WCF service
- Using the Service Model Metadata Utility Tool (Svcutil.exe)
- Sharing contracts with a WCF service and creating a local copy of the service contract
- Adding an endpoint to a client's configuration
- Defining a WCF service contract
- Hosting a service that uses multiple endpoints and modifying a client application to use multiple endpoints
- Hosting a WCF service in Internet Information Server (IIS)
- Defining a data contract and enabling a WCF service to return a custom type to clients
- Creating a strongly typed fault and preventing unhandled exceptions from being converted to a generic exception

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax and have a working knowledge of generic types, attributes, and exception handling in .NET. Although not essential, experience in object-oriented development would also be useful.
Drill: Creating Reliable Services with WCF

Overview
With this Drill, you can work on the foundations of configuration-driven reliable messaging in the Windows Communication Foundation (WCF). Techniques on recovering from data loss, ensuring consistency with transactions, and increasing availability with queues are all included.

Audience
This Drill is for developers who want to be able to use the specific features of WCF that facilitate the development of reliable services running across a distributed environment.

What you'll learn
You will learn the skills needed to create reliable WCF services running across a distributed environment. You will learn how to develop the different types of transactions needed (atomic transactions, Microsoft Message Queuing (MSMQ) transactions, compensating transactions) and you will also be able to implement a reliable session on a custom binding.

Core development tasks
• use atomic transactions in a WCF application
• configure a WCF application to send messages to a queue
• implement MSMQ transactions in WCF
• apply compensating transactions in Windows Communications Foundation WCF
• implement a reliable session on a custom binding in WCF

What you’ll cover
• Using atomic transactions in a WCF application
• Using the TransactionFlow attribute
• Creating a queue on a local machine
• Configuring a WCF application to send messages to, and read messages from, a queue
• Using MSMQ transactions
• Ensuring message queuing is performed in a transactional manner
• Using compensating transactions
• Configuring an application to enable reliable sessions
• Specifying message delivery details

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax. Although not essential, experience in object-oriented development and a familiarity with SQL Server and MSMQ would also be useful.
Drill: Windows Communication Foundation Security

**Overview**
Windows Communication Foundation (WCF) provides a security framework that enables service developers to choose specific implementations of confidentiality, integrity, authentication, and authorization. In this Drill, you perform techniques involving each of these options along with WCF’s auditing support.

**Audience**
This Drill is for developers who want to be able to use the specific features of WCF that can be used to develop the secure solutions needed for services in a distributed environment.

**What you’ll learn**
You will learn key techniques for implementing the secure solutions that are needed in the distributed environment where WCF applications are deployed. These include encryption at transport level, authentication techniques using certificates or username/password combinations, authorization, and impersonation. You’ll also learn to audit security events.

**Core development tasks**
- secure a WCF service by encrypting all communication at the transport level
- create a custom username and password validator to authenticate users of the service
- authenticate the client and service with each other using X.509 certificates
- use WCF’s claims-based Identity Model to authorize users of a service
- configure a WCF service and client applications to impersonate the user currently logged in
- set up security auditing and message logging for a WCF service

**What you’ll cover**
- Using HTTPS to encrypt communication at the transport level
- Setting service configuration files to enable transport-level security
- Creating a custom validator derived from UserNamePasswordValidator
- Configuring a WCF client and service to use message-level security
- Installing and using X.509 certificates to authenticate both client and service
- Authorizing users with WCF’s claims-based Identity Model
- Configuring a WCF service and client to impersonate the user currently logged in
- Implementing both declarative and imperative models for impersonation
- Setting the impersonation level and configuring a service to audit security events
- Using WCF message logging to log and view incoming and outgoing messages

**Prerequisites**
We assume that you’re already familiar with the concepts of programming and you’ve some experience with Visual Studio. For this particular Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax, and have a basic understanding of XML and WCF services. Although not essential, a familiarity with the concepts of public key infrastructure (PKI), certificates, and encryption would also be useful.
Programming WPF using Visual Studio

This Developer Program contains the following Drills:

- Basic Controls and Layout
- Styles and Control Templates
- Data Binding
- Graphics and Animation
- UserControls and Custom Controls with WPF
Drill: Basic Controls and Layout

Overview
This Drill examines basic controls and layout in Windows Presentation Foundation (WPF). Topics include an introduction to WPF controls, using brushes in WPF, creating a user interface in Extensible Application Markup Language (XAML), as well as using StackPanel and WrapPanel. You'll also learn how to create layouts using DockPanel and GridPanel.

Audience
This Drill is for developers who want to gain knowledge of techniques for basic controls and layout in WPF.

What you'll learn
You will learn how to use brushes in WPF, creating a user interface in XAML, as well as using StackPanel and WrapPanel. You'll also learn how to create layouts using DockPanel and GridPanel.

Core development tasks
- explore an introduction to WPF controls
- use brushes in WPF
- create a user interface in XAML
- use StackPanel and WrapPanel
- create layouts using DockPanel and GridPanel

What you'll cover
- Creating WPF user interface controls imperatively
- Painting the background of a window with WPF brushes
- Converting the imperative code of a WPF application to XAML code
- Interacting with objects and properties through XAML
- Refactoring a working application using the StackPanel and WrapPanel layout panels
- Creating a basic user interface layout using the DockPanel control
- Creating an automatic layout using the Grid panel
- Using the Margin property to position controls within a Grid cell

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio. You’ll also need to be comfortable with Visual C# or Visual Basic syntax, and have a basic understanding of XML.
Drill: Styles and Control Templates

Overview
Windows Presentation Foundation (WPF) provides a high degree of control over many aspects of the user interface. This Drill shows different ways to access resources in WPF, how to create styles and use a StyleSelector to apply them, and how to change a control's appearance with a template. It also demonstrates how to create theme-specific styles and templates.

Audience
This Drill is for developers who want to gain a high degree of control over many aspects of the user interface in WPF.

What you’ll learn
You will learn how to create styles and build a StyleSelector to apply them. You’ll also learn how to change a control’s appearance with a template, as well as how to create theme-specific styles and templates.

Core development tasks
- access resources programmatically and declaratively in WPF
- create styles as resources and apply them to user interface elements
- create a StyleSelector that applies a style based on custom logic
- change the visual appearance of an existing control using a control template
- respect the properties of the templated parent in a ControlTemplate using template binding
- create theme-specific styles and templates for custom controls

What you'll cover
- Working with binary and logical resources
- Using styles in WPF to define common property values for elements or controls
- Creating and using a StyleSelector to programmatically apply a style to items in the ItemControl control
- Defining a ControlTemplate for a control to alter its appearance
- Using template binding to perform relative binding within a template
- Creating theme-specific styles and templates for custom controls

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio. You’ll also need to be comfortable with Visual C# or Visual Basic syntax, object-oriented development, and have a basic understanding of XML.
Drill: Data Binding

Overview
Windows Presentation Foundation (WPF) provides many ways to bind user interface elements to various data sources. This Drill shows how to access properties of custom objects and collections using direct binding, how to use DataTemplate objects, and the IValueConverter and IMultiValueConverter interfaces. It also shows how to create a master/detail window that displays ADO.NET data.

Audience
This Drill is for developers who want to gain knowledge in techniques to bind user interface elements to various data sources in WPF.

What you'll learn
You will learn how to access properties of custom objects and collections using direct binding, how to use DataTemplate objects, and how to use the IValueConverter and IMultiValueConverter interfaces. You'll also learn how to create a master/detail window that displays ADO.NET data.

Core development tasks
- bind WPF UI elements to properties of custom objects and collections using direct binding and the CollectionViewSource object
- use INotifyPropertyChanged and INotifyCollectionChanged in a WPF project and explore their effects on the User Interface
- use DataTemplate objects to construct a complex UI for items in a collection
- use the IValueConverter and IMultiValueConverter interfaces to create objects which transform data during data binding
- bind to XML documents using the XmlDataProvide, and to construct, bind, and execute methods on a custom object in XAML using the ObjectDataProvider
- create a master/detail window that displays ADO.NET data using WPF, and to enable this information to be sorted

What you'll cover
- Binding WPF UI elements to collections and classes
- Setting up change notification between a collection of custom CLR objects and the UI
- Displaying multiple properties of a complex CLR object as a single display item in a ListBox
- Transforming values of data-bound controls using custom converter classes which implement the IValueConverter and IMultiValueConverter interfaces
- Constructing and wrapping .NET types in XAML using an ObjectDataProvider, and calling and passing arguments to a method on the wrapped instance
- Binding to database data in WPF

Prerequisites
We assume that you're already familiar with the concepts of programming and you've some experience of Visual Studio. You'll also need to be comfortable with Visual C#, and have a basic understanding of XML and ADO.NET.
Drill: Graphics and Animation

Overview
Windows Presentation Foundation (WPF) provides powerful graphics and animation capabilities. This Drill shows how to apply 2-D transformations to objects, draw 2-D shapes and animate them. It also shows how to create and manipulate elements in a 3-D space, and apply animations to them.

Audience
This Drill is for developers who want to use WPF’s powerful graphics and animation capabilities to create, manipulate, and animate 2-D and 3-D objects.

What you’ll learn
You’ll learn how to apply 2-D transformations to objects, and to draw and manipulate 2-D shapes. You’ll also learn how to manipulate elements in a 3-D environment, and create interactive and animated objects within such an environment.

Core development tasks
- create a reflection effect on a splash screen and its contents by applying 2-D transformations to it
- use Windows Presentation Foundation to draw and describe the geometry of 2-D shapes
- apply WPF animations to 2-D visual elements
- manipulate elements within a WPF 3-D environment
- create interactive 3-D scenes with WPF
- apply animations to objects in a 3-D space

What you'll cover
- Applying rotation and reflection effects
- Hiding a window's non-client area
- Using the Drawing and Geometry classes
- Using the Path element with Path Markup syntax
- Using animation timelines
- Adding an EventTrigger element
- Building a 3-D scene with XAML markup
- Adding a camera and light source
- Implementing hit testing in a 3-D UI
- Using the Viewport3D element
- Using DoubleAnimation
- Using AxisAngleRotation3D rotations and TranslateTransform3D transformations

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio. You'll need to be comfortable with Visual C# or Visual Basic syntax, object-oriented development, and have a basic understanding of XML. You'll also need to be familiar with 2-D and 3-D basic concepts.
Drill: UserControls and Custom Controls with WPF

Overview
This Drill shows how to create reusable controls in Windows Presentation Foundation (WPF). Topics include user controls, dependency properties and routed events, named template parts and attached properties, and commands.

Audience
This Drill is for developers who want to gain knowledge of techniques for creating reusable controls in WPF.

What you'll learn
You'll learn how to create a DependencyProperty and RoutedEvent in a UserControl, as well as how to create and use custom controls and lookless custom controls. You'll also learn how to use attached properties and named template parts in a custom control, how to design a single-child element based on FrameworkElement, and how to use commands.

Core development tasks
• create and use a custom UserControl in WPF
• increase code reuse and improve code quality with user controls
• add a DependencyProperty and a RoutedEvent to a UserControl
• implement a custom control in WPF with a default style that can be overridden by the application that uses the control
• apply custom commands to user interface elements
• declare and register an attached property and use a named template part
• design a single-child element based on the FrameworkElement class and adapt its rendering behavior

What you'll cover
• Encapsulating reusable code and providing defined styles for specific data types
• Creating and registering a custom dependency property and routed event in WPF
• Using routed events with dependency properties
• Defining custom commands and binding handlers to the CanExecute and Executed events
• Binding keyboard shortcuts to the commands of application controls
• Creating and registering a custom attached dependency property
• Storing the content of an element in a specific property when the FrameworkElement instance is used in XAML
• Sizing and positioning using the WPF graphic system and interacting with the DrawingContext in the OnRender method

Prerequisites
We assume that you're already familiar with the concepts of programming and you've some experience of Visual Studio. You'll also need to be comfortable with Visual C# or Visual Basic syntax, and object-oriented development. A basic understanding of WPF concepts and XAML is also desirable.
Programming WF using Visual Studio

This Developer Program contains the following Drills:

- Programming Windows Workflow Foundation Fundamentals
- Programming Sequential Workflows
- Programming State Machine Workflows
- Developing Custom Activities
- Programming Runtime Services and Hosts
Drill: Windows Workflow Foundation Fundamentals

Overview
This Drill introduces Windows Workflow Foundation (WF). It lets you examine basic techniques useful for developing and running both sequential and state-based workflows, including creating and consuming simple activities.

Audience
This Drill is for developers who want to use WF—a key component of .NET Framework 3.0—to quickly build workflow-enabled applications.

What you'll learn
You will learn how to quickly build workflow-enabled applications using Microsoft's new WF technology. The skills you will learn include creating sequential and state-based workflows, flow control, passing parameters from a host, listening to events, and creating a tracking service.

Core development tasks
- create a basic sequential workflow and use the activities DelayActivity and CodeActivity
- employ the basic ElseIfActivity and WhileActivity flow-control activities, and set up properties using the rule engine
- use and pass input parameters to a sequential workflow
- apply a ListenActivity activity in a sequential workflow to handle external events from a host application
- implement a state-based workflow and handle external events using EventDrivenActivity
- track the executing state of a sequential workflow using a tracking service and raise application-specific tracking data from the workflow

What you'll cover
- The basic activities CodeActivity and DelayActivity
- Attaching event handlers to activities to perform custom business logic
- The flow-control activities ElseIfActivity and WhileActivity
- Passing parameters from a host application to a sequential workflow
- Listening to events
- Pausing a workflow to await user input
- Creating a state-based workflow
- Processing tasks from a host application through different states in a workflow
- Adding a pre-built TrackingService service to a workflow runtime
- Using the TrackData method to send user tracking data to a tracking service

Prerequisites
We assume that you're already familiar with the concepts of programming and you've some experience with Visual Studio. For this particular Drill, you'll also need to be comfortable with Visual C# or Visual Basic syntax, and have a basic understanding of workflows. Although not essential, experience in object-oriented development would also be useful.
Drill: Programming Sequential Workflows

Overview
This Drill lets you practice basic techniques on developing and running sequential workflows. Topics cover building and consuming custom activities, using compensation handlers, using the ReplicatorActivity, dynamic updating, and policy.

Audience
This Drill is for developers who want to gain knowledge of techniques useful for developing and running sequential workflows.

What you’ll learn
You will learn how to build and consume custom activities. You will also learn about using compensation handlers, using the ReplicatorActivity, dynamic updating, and policy.

Core development tasks
- extend the SequenceActivity class to encapsulate workflow logic in a custom activity
- use compensation in workflows
- employ the Replicator Activity
- modify a workflow at runtime
- define rules using the Policy Activity
- host the Workflow Designer and monitor a workflow

What you’ll cover
- Encapsulating workflow logic in a custom activity
- Using compensation in workflows
- Creating an arbitrary number of instances of a single activity during run time
- Using the dynamic workflow capabilities of Windows Workflow Foundation
- Adding a new activity to a running workflow
- Using the rules engine capabilities in Windows Workflow Foundation
- Declarative modelling of application logic units in a business process
- Hosting the workflow designer within a Windows Forms application
- Displaying the progress of a running workflow instance

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio, including its Workflow Designer. For this Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax, working with workflows and activities, and object-oriented development. Although not essential, experience with Windows Forms applications would also be useful.
Drill: Programming State Machine Workflows

Overview
This Drill examines basic techniques useful for developing and running state machine workflows. Topics include building and consuming Local Services for communication between the workflow and host, using workflow queue data to determine the available actions for a particular state, and using role-based authorization.

Audience
This Drill is for developers who want to gain knowledge of techniques useful for developing and running state machine workflows.

What you’ll learn
You will learn how to build and consume Local Services for communication between the workflow and host. You will also learn how to use workflow queue data to determine the available actions for a particular state, and to use role-based authorization.

Core development tasks
- define states and transitions in a state machine workflow
- execute logic when the state machine workflow enters or exits a state
- use Local Services to communicate between the host and a state machine workflow
- define correlation in a state machine workflow
- manage workflow queue data
- implement roles in a state machine workflow

What you’ll cover
- Defining states and transitions in a state machine workflow
- Managing transitions between states based on user input
- Sharing activities between states using a composite state
- Executing logic when the state machine workflow enters or exits a state
- Adding state initialization and state finalization activities to a state machine workflow
- Defining and using Local Services to communicate between the host and a state machine workflow
- Defining correlation in a state machine workflow
- Managing workflow queue data
- Using roles with workflow activities to restrict access from within the workflow

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio, including its Workflow Designer. For this Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax, working with workflows and activities, and object-oriented development.

Drill Details

<table>
<thead>
<tr>
<th>Status</th>
<th>Released</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Program</td>
<td>Programming WF using Visual Studio</td>
</tr>
<tr>
<td>Duration</td>
<td>3 hours</td>
</tr>
<tr>
<td>Programming Language</td>
<td>C# &amp; VB</td>
</tr>
<tr>
<td>System Requirements</td>
<td>A non-express edition of Visual Studio</td>
</tr>
<tr>
<td></td>
<td>Microsoft .NET Framework 3.0 Redistributable Package</td>
</tr>
<tr>
<td></td>
<td>Visual Studio 2005 extensions for .NET Framework 3.0 (Windows Workflow Foundation)</td>
</tr>
<tr>
<td>Level</td>
<td>2</td>
</tr>
<tr>
<td>Drill Code</td>
<td>D0042B-CS D0042B-VB</td>
</tr>
</tbody>
</table>
Drill: Developing Custom Activities

Overview
This Drill covers the development of custom workflow activities. Topics include creating a simple custom activity, and attaching to it a custom designer and toolbox item class. Other topics include enhancing custom activities with validation and compensation logic, and grouping child activities into a composite sequential activity.

Audience
This Drill is for developers who want to develop applications with custom workflow activities ranging from simple examples to composite sequential activities, and which feature enhancements such as validation and compensation.

What you’ll learn
You will learn how to create a simple custom activity and attach to it a custom designer and toolbox item class. You'll also learn the skills needed to enhance custom activities with validation and compensation logic, and group child activities into a composite sequential activity.

Core development tasks
- create a simple custom activity
- add custom validation to a custom activity
- add designer and toolbox support to custom activity
- create a custom composite sequential activity
- manage child activities in a custom composite activity
- implement cancellation and compensation in custom composite activities

What you'll cover
- Creating a simple custom activity that derives from the Activity class
- Using a custom activity in a sequential workflow
- Implementing a custom designer
- Implementing a custom toolbox item class
- Attaching a custom designer and a custom toolbox item to a simple custom activity
- Adding validation to a custom activity
- Using default Windows Workflow validation with a custom validator class
- Grouping child activities into a composite sequential activity
- Managing execution of child activities within a custom composite activity
- Implementing compensation logic
- Overriding default cancellation behavior of custom composite activities

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve experience with the Visual Studio Environment, including its Workflow Designer. For this Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax, working with workflows and activities, and object-oriented development.
Drill: Programming Runtime Services and Hosts

Overview
This Drill examines basic techniques useful for developing runtime services and hosts. Topics include hosting a workflow in a Web application, publishing a workflow as a Web service, persisting workflow instances with SqlWorkflowPersistenceService, and using SqlTrackingService. Other topics include creating and using a custom tracking service and tracking channel, and creating transactional services.

Audience
This Drill is for developers who want to learn the techniques for programming runtime services and hosts in a Windows Workflow Foundation environment.

What you’ll learn
You will learn how to host a workflow in a web application and publish a workflow as a web service. You’ll also learn how to persist workflow instances and create transactional services, and how to implement various techniques for tracking workflow information.

Core development tasks
• host a workflow in an ASP.NET web application
• publish a workflow as a web service
• persist workflow instances using SqlWorkflowPersistenceService
• track workflow information using SqlTrackingService
• create and use a custom tracking service and tracking channel
• create transactional services

What you’ll cover
• Hosting a workflow in an ASP.NET web application
• Configuring a web application workflow
• Publishing a workflow as a web service
• Persisting workflow state to a SQL Server database
• Using SqlWorkflowPersistenceService
• Tracking workflow information using SqlTrackingService
• Adding a custom tracking service to a workflow
• Defining a custom track point which specifies the data to extract from a workflow
• Creating transactional services
• Implementing the IPendingWork interface

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve experience with the Visual Studio Environment, including its Workflow Designer. You’ll also need to be comfortable with Visual C# or Visual Basic syntax, working with workflows and activities, web applications and services, and object-oriented development.
Programming with VSTS using Visual Studio

This Developer Program contains the following Drills:

- Unit Testing with Visual Studio Team System
- Static Analysis, Profiling & Code Coverage with Visual Studio Team System
Drill: Unit Testing with Visual Studio Team System

Overview
This Drill focuses on effective techniques and patterns that developers can apply while creating and modifying test classes. Tests are implemented using the unit testing framework included in Visual Studio Team System (VSTS).

Audience
This Drill is for developers who want to learn how to write effective unit tests using documented xUnit patterns in Visual Studio Team System.

What you'll learn
You will learn how to use the Unit Testing Framework in Visual Studio Team System to create unit tests for your code. After taking this Drill, you will be able to use Team System to create unit tests on a class or method level, access private methods and properties of your code under test, and use the Test View window in Visual Studio to control which tests run. You will also know how to apply documented xUnit testing patterns and techniques.

Core development tasks
- create unit tests using the Unit Testing Framework in VSTS
- refactor existing code using the Dependency Injection pattern to increase testability
- control inputs to your system under testing using Test Stubs
- verify a component's behavior by replacing objects with Mock objects
- increase the performance of tests by using Shared Fixtures
- reduce duplication in test code by using Custom Assertion and Finder utility methods

What you'll cover
- Using the Unit Testing Framework in VSTS & using Private Accessors to test private methods from unit testing code
- Implementing system-under-test functionality with Test Stubs and Mock objects
- Increasing test performance by using Shared Fixtures
- Improving testability by using Dependency Injection and reducing test code duplication by using testing utilities

Prerequisites
We assume that you're already familiar with the concepts of programming and you've some experience of Visual Studio. For this Drill, you'll also need to be comfortable with Visual C# or Visual Basic syntax.
Drill: Static Analysis, Profiling, and Code Coverage with Visual Studio Team System

Overview
This Drill focuses on using the Code Coverage, Profiling, and Static Analysis tools included in Visual Studio Team System (VSTS).

Audience
This Drill is for developers who want to learn how to identify performance bottlenecks, incomplete unit tests, and non-conforming code in their applications using the tools included in Visual Studio Team System.

What you'll learn
You will learn how to enable the Code Coverage features in VSTS that allow you to visually identify paths through your source code that are not tested by unit tests. You will also learn how to gather performance data from applications and use this performance data to optimize your code and remove bottlenecks.

In this Drill, you will also find out how to examine your source code using the Static Analysis Tools in Visual Studio Team System. Afterwards, you will be able to identify non-uniform and potentially hazardous code using the built-in rule. In addition, you’ll know how to create custom static analysis rules.

Core development tasks
- examine your unit tests and discover untested code
- identify performance bottlenecks in your applications
- analyze your code using the built-in static analysis rules provided by VSTS
- create custom static analysis rules to identify non-compliant code

What you’ll cover
- Using Code Coverage tools in Visual Studio Team System
- Analyzing performance reports to identify bottlenecks in code
- Manually adding a static analysis rule and using it to identify and resolve non-compliant code
- Creating a custom static analysis rule

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio and its unit testing features. For this Drill, you’ll also need to be comfortable with Visual C# or Visual Basic syntax.

Drill Details
Status
Released

Developer Program
Programming with Visual Studio Team System using Visual Studio

Duration
2.5 hours

Programming Language
C# & VB

System Requirements
- A non-express edition of Visual Studio

Level
3

Drill Code
- D0044B-CS
- D0044B-VB
Transact-SQL Development using Visual Studio

This Developer Program contains the following Drills:

• Transact-SQL Fundamentals in SQL Server
• Transact-SQL Enhancements in SQL Server
Drill: Transact-SQL Fundamentals in SQL Server

Overview
This Drill covers fundamental T-SQL activities such as making queries, sorting and filtering, and using table joins. It also covers control-of-flow language, and using functions (including aggregate functions) to manipulate data.

Audience
This Drill is for developers who want to learn the skills to use Transact-SQL to interact with a SQL Server database.

What you'll learn
You will learn the basic skills needed to use the T-SQL language to access Microsoft's popular SQL Server database. This includes making queries, sorting and filtering data, and using table joins. You'll also cover control-of-flow language, and using functions (including aggregate functions) to manipulate data.

Core development tasks
- Make basic T-SQL queries
- Sort and filter in T-SQL
- Use table joins and column aliases
- Use aggregate functions with GROUP BY and HAVING
- Use T-SQL control-of-flow language
- Manipulate data with built-in T-SQL functions

What you'll cover
- Using the basic T-SQL statements SELECT, UPDATE, INSERT, and DELETE
- Sorting and filtering data using T-SQL
- Combining records from two tables
- Using different types of joins
- Changing the name of a result set column
- Performing a calculation on a set of values
- Returning a single value using an aggregate function
- Controlling the flow of execution of T-SQL statements
- Using built-in string and system scalar functions

Prerequisites
We assume that you're already familiar with programming concepts and you've used Visual Studio. You'll also need to be comfortable with Visual C# or Visual Basic syntax, and have a working knowledge of SQL and SQL Server (or general database fundamentals, at least). Although not essential, experience with Transact-SQL (T-SQL) would be useful.
Drill: Transact-SQL Enhancements in SQL Server

Overview
This Drill covers advanced T-SQL activities such as subqueries and Set operations, table expressions, transactions, and structured exception handling. It also covers how to group, rank, summarize, and filter data using T-SQL, as well as how to query relational and XML data using FOR XML and XQUERY.

Audience
This Drill is for developers who want to use new Transact-SQL capabilities in SQL Server 2005.

What you'll learn
You'll learn the skills needed to use the enhanced T-SQL capabilities available in SQL Server 2005. This includes subqueries and set operations, table expressions, transactions, and structured exception handling. You'll also learn how to group, rank, summarize, and filter data using T-SQL, as well as how to query relational and XML data using FOR XML and XQUERY.

Core development tasks
- Use subqueries and set operations
- Use table expressions
- Manage transactions
- Implement exception handling
- Group, rank, summarize and filter in T-SQL
- Query relational and XML data, using FOR XML and the XML data type

What you'll cover
- Creating T-SQL statements that contain nested queries
- Selecting rows from two result tables based on set operations
- Writing maintainable SQL statements with derived tables, table expressions, and recursion
- Managing database transactions in SQL Server 2005
- Raising and handling exceptions in SQL Server 2005
- Grouping, ranking, summarizing, and filtering data
- Using T-SQL's ranking and pivoting functions
- Querying data stored using the XML data type in SQL Server 2005
- Querying relational data

What you'll need to know
We assume that you're already familiar with programming concepts and you've used Visual Studio. You'll also need to be comfortable with Visual C# or Visual Basic syntax, and have a working knowledge of SQ and SQL Server. Although not essential, experience with Transact-SQL (T-SQL) and SQL Server 2005 would be useful.
Silverlight 3 Programming using Visual Studio

This Developer Program contains the following Drills:

- Managed Code in Silverlight 3
- Working with Data in Silverlight 3
Drill: Managed Code in Silverlight 3

Overview
This Drill covers important managed code techniques in Silverlight 3, Microsoft's cross-platform technology for developing rich, interactive applications on the Web.

Audience
This Drill is for developers who want to learn how to use managed code techniques to develop and deploy Silverlight applications.

What you'll learn
You'll learn how to use the Silverlight DataGrid and MediaPlayer ASP.NET controls, how to use Silverlight Panels to create an Application Layout, and how to handle events and to interact with Web pages.

Core development tasks

- Deploy a Silverlight application
- Configure Silverlight libraries for in-package or on-demand deployment
- Arrange a Web application using Silverlight Panel controls
- Customize the Silverlight DataGrid control
- Wire UI events in XAML
- Work with routed events
- Attach managed handlers to events from HTML elements on the Silverlight hosting page
- Make methods of managed objects available to JavaScript
- Modify elements of a host HTML page from Silverlight

What you'll cover

- Hosting an application in an ASPX page
- Personalizing the look of the MediaPlayer control
- Arranging large elements of an application UI using the Grid panel
- Using the Silverlight DataGrid control
- Changing the default columns by using the Columns collection of the DataGrid
- Using the built-in column type DatGridTextColumn
- Creating a custom template that includes the DatePicker control
- Wiring up Silverlight events in XAML
- Handling Silverlight events with managed code
- Handling routed events and determining the objects that raise routed events
- Calling managed code from JavaScript
- Modifying values of a Web page's DOM from managed Silverlight code

Prerequisites
We assume that you're already familiar with programming concepts and that you've used Visual Studio. You'll also need to be comfortable with writing C#, HTML, XAML, and JavaScript.
Drill: Working with Data in Silverlight 3

Overview
This Drill covers important techniques for working with data in Silverlight 3, Microsoft's cross-platform technology for developing rich, interactive applications on the Web.

Audience
This Drill is for developers who want to learn how work with data Silverlight 3.

What you'll learn
You'll learn how to use Silverlight 3 techniques such as data binding, implementing a master detail view, accessing HTTP-based services, accessing SOAP services, accessing syndication feeds, and using isolated storage.

Core development tasks
- Perform data binding in Silverlight 3
- Implement one-way data binding on a custom class
- Implement a Master Detail view
- Use a custom data template to improve the look and feel of the Silverlight ListBox control
- Interact with RESTful services over HTTP using XML
- Access data from a WCF service
- Convert retrieved data values using the IValueConverter interface
- Consume an RSS feed and display the consumed data in the DataGrid control
- Manage local assets by accessing Silverlight's isolated storage and application settings

What you'll cover
- Binding simple data values to XAML elements
- Modifying one class to support data binding
- Creating your own ItemTemplates for ListBox items
- Calling HTTP services from a Silverlight 3 application
- Using the Webclient to set up asynchronous calls to remote, HTTP based services
- Making WCF services accessible to Silverlight 3 clients by using the correct binding type
- Creating a proxy for Web services using Visual Studio
- Solving cross-domain issues in Silverlight 3 by placing correctly-configured clientaccesspolicy.xml or crossdomain.xml files on the required Web server
- Using the System.ServiceModel.Syndication namespace for RSS and Atom feeds
- Rendering the DataGrid control on screen
- Creating, saving, and deleting files in Silverlight 3's isolated storage virtual file system
- Using the IsolatedStorageSettings static class

Prerequisites
We assume that you're already familiar with programming concepts and that you've used Visual Studio. You'll also need to be comfortable with writing C#, HTML, XAML, and JavaScript.
.NET Framework 3.5 Programming using Visual Studio

This Developer Program contains the following Drills:

- New features in C# 3.0/VB 9.0
- LINQ to SQL
- LINQ to XML
Drill: New features in C# 3.0/VB 9.0

Overview
The Drill covers important new features in C# 3.0/VB 9.0.

Audience
This Drill is for developers who want to be able to use the new features in C# 3.0/VB 9.0.

What you’ll learn
You will learn important new C# 3.0/VB 9.0 features, including delegates and lambda expressions, as well as basic and advanced LINQ queries.

Core development tasks
- define, instantiate, and use delegates
- create and use extension methods
- use the auto-implemented properties language feature as shorthand for property declaration
- create basic lambda expressions and Func delegates
- manipulate collections of data using new LINQ extension methods with lambda expressions
- join two sequences
- use partitioning, set, and conversions operators

What you’ll cover
- defining and creating an instance of the anonymous type
- creating objects and collections using short-hand syntax
- creating, importing, and using extension methods
- creating properties using short-hand syntax
- defining argument and return types with the Func delegate
- providing criteria to the extension methods, using lambda expressions
- removing duplicates from a collection using the Distinct method
- performing an inner join between two in-memory sequences
- performing a grouped join between two sequences of objects
- retrieving the common elements from two sequences, using the Intersect set operator
- filtering the elements of a sequence based on a type, using the OfType conversion operator

Prerequisites
We assume that you're already familiar with the concepts of programming and you're familiar with previous versions of the Visual Studio environment. For this particular drill, you'll also need to be comfortable with the generics, iterators, and delegates features of C# 2.0/VB 8.0.
Drill: LINQ to SQL

Overview
LINQ to SQL provides a means for managing relational data as objects while retaining the ability to query that data. This Drill covers a number of techniques needed to use LINQ to SQL effectively, including retrieving data, CRUD operations, and optimization.

Audience
This Drill is for developers who want to be able to use the LINQ to SQL functionality available in .NET Framework 3.5.

What you'll learn
You'll be able to use LINQ to SQL to manage relational data as objects, and perform database operations through a LINQ to SQL data context. You'll also be able to optimize LINQ to SQL, and use it against existing stored procedures.

Core development tasks
- generate strongly typed LINQ to SQL data context and entity classes to retrieve data from a database
- create, retrieve, update, and delete (CRUD) database data through a LINQ to SQL data context
- use LINQ to SQL entity classes against existing stored procedures
- detect and handle concurrency conflicts
- configure LINQ to SQL at runtime, to load data and execute queries more efficiently
- use the LinqDataSource control

What you'll cover
- generating strongly typed LINQ to SQL data context and entity classes
- using the ORM designer
- using LINQ to SQL data context and entity classes to retrieve data from a database
- optimizing LINQ to SQL queries
- implementing compiled queries
- using LinqDataSource to bind ASP.NET UI controls to database data
- enabling concurrency checks
- parsing the elements of change conflicts
- using LINQ to SQL entity classes against existing stored procedures

Prerequisites
We assume that you're already familiar with the concepts of programming and with previous versions of the Visual Studio environment. For this particular drill, you'll also need to be comfortable with Visual Studio, as well as SQL Server. Although not essential, knowledge of data binding techniques in Windows or web applications would also be useful.
Drill: LINQ to XML

Overview
LINQ to XML provides a clean programming model that enables you to read, construct, and write XML data. This Drill covers a number of important techniques required to use LINQ to XML effectively.

Audience
This Drill is for developers who want to be able to use the LINQ to XML functionality available in .NET Framework 3.5.

What you'll learn
After taking this Drill, you’ll be able to use LINQ to XML to read, write, and process XML data.

Core development tasks
- read and write data
- work with advanced XML document settings
- navigate an XML tree
- update XML
- validate XML with extension methods
- query and transform XML with XPath and XSLT

What you’ll cover
- Creating, loading, and saving XML documents and elements using LINQ to XML
- Working with the XDocument class provided by LINQ to XML
- Using LINQ to XML to specify settings such as declarations, processing instructions, and namespaces
- Creating a namespace with a prefix using both the XNamespace class and its XmlNs property
- Navigating through and locate various nodes in XML using LINQ To XML
- Modifying and updating in-memory XML with methods of the XNode and XElement classes
- Loading an XML Schema Definition (XSD) file using types from the pre-LINQ System.Xml namespace
- Using LINQ to XML to locate nodes in an XML file with XPath expressions
- Using the "bridge classes" provided by LINQ to XML
- Transforming an XML file to an alternative representation using XSLT

Prerequisites
We assume that you're already familiar with the concepts of programming and with previous versions of the Visual Studio environment. For this particular drill, you’ll also need to be comfortable with Visual C#, and have a working knowledge of XML syntax. Although not essential, knowledge of XSD, XSLT and XPath would also be useful.
Unit Testing with Visual Studio

This Developer Program contains the following Drills:

- Unit Testing with Visual Studio
- Unit Testing with NUnit
- Unit Testing with xUnit.net
Drill: Unit Testing with Visual Studio

**Audience**
This Drill is for intermediate developers who want to learn how to write effective unit tests using documented test patterns in Visual Studio.

**What you’ll learn**
You will learn how to use the Unit Testing Framework in Visual Studio to create unit tests for your code. After taking this Drill, you will be able to use Visual Studio to create unit tests on a class or method level, access private methods and properties of your code under test, and use the Test View window in Visual Studio to control running tests. You will also learn how to apply documented testing patterns and techniques.

**Core development tasks**
- create unit tests using the Unit Testing Framework in Visual Studio
- refactor existing code using the dependency injection pattern to increase testability
- control inputs to your system under testing using test stubs
- verify a component’s behavior by replacing objects with mock objects
- increase the performance of tests by using shared fixtures
- reduce duplication in test code by using custom assertion and finder utility methods
- use private accessors to test private methods from unit testing code
- implement system-under-test functionality with test stubs and mock objects
- improve testability by using dependency injection and reduce test code duplication by using testing utilities

**Prerequisites**
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio. For this Drill, you’ll also need to be comfortable with Visual C# syntax.
Drill: Unit Testing with NUnit

Audience
This Drill is for intermediate level developers who want to learn how to write effective unit tests using documented test patterns in NUnit, the open source unit testing framework.

What you'll learn
You will learn how to use the NUnit framework to create unit tests for your code. After taking this Drill, you will be able to use NUnit to create unit tests on a class or method level and control the tests that run. You will also know how to apply documented testing patterns and techniques.

Core development tasks
- create unit tests using the NUnit framework
- refactor existing code using the dependency injection pattern to increase testability
- control inputs to your system under testing using test stubs
- verify a component's behavior by replacing objects with mock objects
- increase the performance of tests by using shared fixtures
- reduce duplication in test code by using custom assertion and finder utility methods
- implement system-under-test functionality with test stubs and mock objects
- improve testability by using dependency injection and reduce test code duplication by using testing utilities

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio. For this Drill, you’ll also need to be comfortable with Visual C# syntax.

Drill Details
Status
Released

Developer Program
Unit Testing with Visual Studio

Number of Tasks
6

Duration
3 hours

Programming Language
C# & VB

Level
3

Drill Code
- D0057-CS
- D0057-VB
Drill: Unit Testing with xUnit.net

Audience
This Drill is for intermediate developers who want to learn how to write effective unit tests using documented test patterns in xUnit.net, the open source unit testing framework.

What you'll learn
You will learn how to use the xUnit.net framework to create unit tests for your code. After taking this Drill, you will be able to use xUnit.net to create unit tests on a class or method level, access private methods and properties of your code under test, and control the tests that run. You will also know how to apply documented testing patterns and techniques.

Core development tasks
• create unit tests using the xUnit.net framework
• refactor existing code using the dependency injection pattern to increase testability
• control inputs to your system under testing using test stubs
• verify a component's behavior by replacing objects with mock objects
• increase the performance of tests by using shared fixtures
• reduce duplication in test code by using custom assertion and finder utility methods
• use private accessors to test private methods from unit testing code
• implement system-under-test functionality with test stubs and mock objects
• improve testability by using dependency injection and reduce test code duplication by using testing utilities

Prerequisites
We assume that you’re already familiar with the concepts of programming and you’ve some experience of Visual Studio. For this Drill, you’ll also need to be comfortable with Visual C# syntax.