Introduction to Visual Studio Technologies

This book is included in this form by Microsoft Corporation by arrangement and through permission from Apress Media.

Apress®

Apress Media

Introduction to Visual Studio Technologies

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher of Apress: Paul Manning
Lead Editor: Jeffrey Pepper

Coordinating Editor: Chris Nelson
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `www.springeronline.com`.

For information on translations, please e-mail `rights@apress.com`, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. Ebook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at `www.apress.com`. For detailed information about how to locate your book’s source code, go to www.apress.com/source-code

Contents at a Glance

Related Titles from Apress

Apress titles are available in print and electronic form at computer booksellers and electronic bookstores around the world. Many of the titles listed here are available or will be available soon on Apress.com in the Apress Alpha book program. Therein you can purchase a book, get chapters as they are developed, and in the end get the complete final ebook – all for the price of a normal Apress ebook. It's a great way to get started as our authors create the books. But before you do, go to the back of this ebook to get the 30% OFF COUPON CODE. Thereafter, simply go to the book's page and click the yellow *Buy Alpha Book* button, enter the code and follow the instructions. Y can go back to Apress.com at any time and download the latest additions to your ebook. Apress will notify you upon publication when the final ebook is available.

Upcoming publications include:

Available in May:

Metro Revealed: Building Windows 8 apps with XAML and C#

Available in June:

Pro C# and the .NET 4.5 Framework, 6th Edition

Available this spring:

WinRT Revealed

Pro WinRT using C# and XAML
978-1-4302-4515-5

Beginning Silverlight 5 in C#, 4th Edition

Available this summer:

Metro Style Application Recipes for Windows 8 in C#

Pro Windows 8 Development with XAML and C#

Pro Application Lifecycle Management with Visual Studio, 2nd Edition

Pro HTML5 Performance

Beginning HTML5 and CSS3

Foundation HTML5 with CSS3

Pro JavaScript for Web Apps

Beginning ASP.NET 4.5 in C#

Beginning ASP.NET 4.5 in VB

Ultra-Fast ASP.NET 4.5, 2nd Edition

Beginning ASP.NET 4.5 Databases, 3rd Edition

Beginning C# 2012 Databases, 2nd Edition

Illustrated C# 2012, 4th Edition

Introducing .NET 4.5, 2nd Edition

Pro WF 4.5

The Windows 8 Power Users Guide

Pro SQL Azure, 2nd Edition

Available this fall:

Beginning Metro Application Development in Windows 8, XAML Edition
978-1-4302-4566-7

Pro Visual Studio 11

Beginning Windows 8 App Development
978-1-4302-4563-6

Pro HTML5 with Visual Studio 2012
978-1-4302-4638-1

Pro Business Metro Style Apps in XAML

Pro HTML5 Application Development

Pro ASP.NET MVC 4, 4th Edition

Pro ASP.NET 4.5 in C#, 5th Edition

Pro ASP.NET 4.5 in VB, 5th Edition

JavaScript Programmer's Reference
978-1-4302-4629-9

Pro WPF in C# 2012, 4th Edition

Pro Windows 8 Development with HTML5 and JavaScript

Pro .NET Performance

Related published publications include:

Pro Application Lifecycle Management with Visual Studio, 2nd Edition

Windows Azure Platform, 2nd Edition

The Definitive Guide to HTML5

HTML5 Mastery

CSS Mastery, 2nd Edition

DOM Scripting, 2nd Edition

Pro .NET Best Practices

Pro Business Applications with Silverlight 5, 2nd Edition

Pro Silverlight 5 in C#, 4th Edition

Pro Silverlight 5 in VB, 4th Edition

Beginning Kinect Programming with the Microsoft Kinect SDK

Meet the Kinect

Hacking the Kinect

Pro Visual Studio LightSwitch 2011 Development

Pro NuGet

Beginning Windows Phone App Development

Pro ASP.NET MVC 3 Framework, 3rd Edition

Pro LINQ

Pro .NET 4 Parallel Programming in C#

About the Authors

Adam Freeman is an experienced IT professional who has held senior positions in a range of companies, most recently serving as Chief Technology Officer and Chief Operating Officer of a global bank. Now retired, he spends his time writing and running. Adam wrote the first chapter for this project and he wrote Chapter 2 for his book *Metro Revealed: Building Windows 8 apps with XAML and C#*

His other upcoming publications include:

Available in May:

Metro Revealed: Building Windows 8 apps with HTML5 and JavaScript

Metro Revealed: Building Windows 8 apps with XAML and C#

Available this summer:

Pro JavaScript for Web Apps

Available this fall:

Pro Windows 8 Development with HTML5 and JavaScript

Pro Visual Studio 11

Pro ASP.NET MVC 4, 4th Edition

Pro ASP.NET 4.5 in C#, 5th Edition

Pro ASP.NET 4.5 in VB, 5th Edition

His other publications include:

Pro jQuery

The Definitive Guide to HTML5

Applied ASP.NET 4 in Context

Pro ASP.NET MVC 3 Framework, 3rd Edition

Introducing Visual C# 2010

Pro ASP.NET 4 in C# 2010, 4th Edition

Pro ASP.NET 4 in VB 2010, 3rd Edition

Pro LINQ

Pro .NET 4 Parallel Programming in C#

Visual C# 2010 Recipes

Hrishikesh Dewan has been a Consultant for CT DC INDIA TEC in Siemens Information Systems Ltd since June, 2008 where he has had the experience of leading a team of 10 to 15 budding talents a the Microsoft Technologies Competency Center. The goal of this competency center was to utilize Microsoft based technologies: .NET framework, tools and frameworks related to the development of software (mainly in C#) and apply them to create innovative solutions. Of late, the competency center has diverged to create a distributed systems lab where the main focus is on designing algorithms and techniques to crack solutions to large scale computational and storage problems. Prior to joining SISL, he founded a very small open source software development organization named "Srishti" that promoted open source tools and software for schools and colleges. He led a modest team to distribute a localized (in Assamese) GNU Linux distribution and a collaborative e-learning tool. He also worked as a project manager for 2 years at 'Eth Ltd', a subsidiary and R&D unit of Dishnet Wireless Ltd (Aircel). I his 5 years at ETH he was also involved in designing file systems, integrated set top boxes for learning and most importantly on management of hardware and software in high performance clusters and computing networks (optical, coaxial switched computer networks). Srishti started his career as a consultant, designing and developing software for utility companies like gas, power, oil and varied allied sectors. He wrote the testing chapter in this book.

Joachim Rossberg has worked as an information technology consultant since 1998. He is primarily a project manager but has an extensive history as a system developer/designer. He has demonstrated his technical background with various achievements over the years: MCSD, MCDBA, MCSA, and MCSE. His specialties include project management, operating systems, SQL Server, Visual Basic, and Visual Studio. Joachim is now working for Know IT Consulting in Gothenburg, Sweden. The three chapters from this book on ALM are adapted from his book *Pro Application Lifecycle Management with Visual Studio*, 2nd Edition.

His contribution was excerpted from this book:

Pro Application Lifecycle Management with Visual Studio, 2nd Edition

His other publications include:

Pro Scalable .NET 2.0 Application Designs, 2nd Edition

Pro Visual Studio Team System Application Lifecycle Management

Tejashwi Redkar is a senior consultant at Microsoft Consulting Services. He has a master's degree from San Jose State University in California and is an experienced architect of service-oriented systems for financial services, telemetry, and e-commerce. The two chapters in this book on Azure are adapted from his book, *Windows Azure Platform*, 2nd Edition.

His contribution was excerpted for this book:

Windows Azure Platform, 2nd Edition

His other publications include:

Pro .NET 1.1 Remoting, Reflection, and Threading

CHAPTER 1

Introducing Visual Studio

This chapter was submitted by Adam Freeman

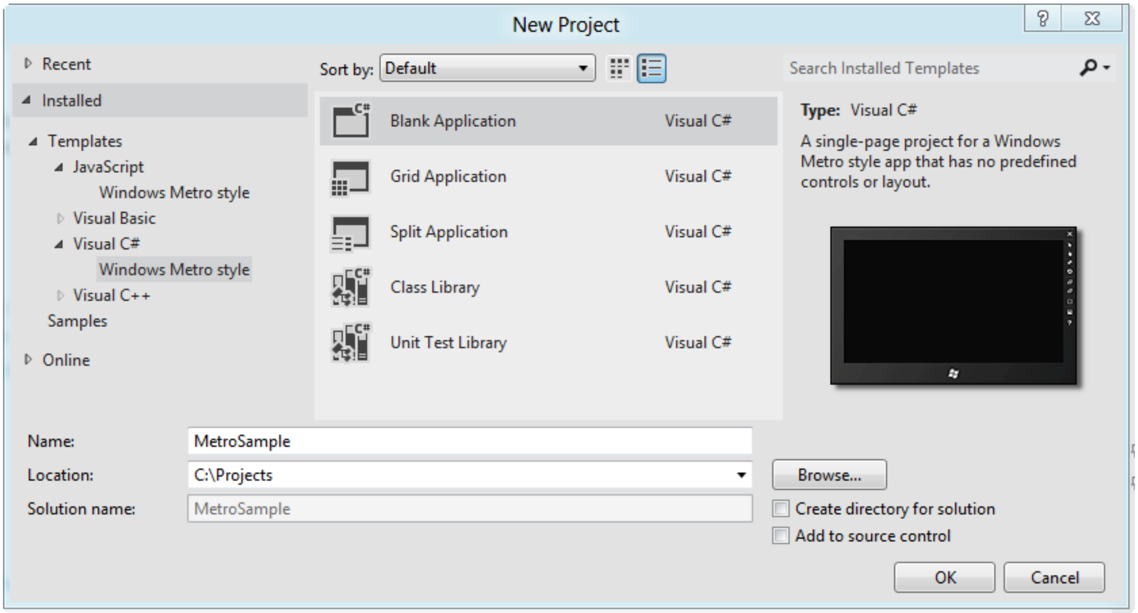
The new Visual Studio is a rich and complex integrated development environment. It supports a range of languages and an even larger range of project types. In this chapter, I will give you a high-level overview of the most important parts of Visual Studio. I start with showing you how to create a project before going on to introduce the code editor, the visual design surface and the all-important build and debugging tools.

I only have space to cover the essentials, so at the end of each section I have listed three features which I suggest you explore further. These are the features that I find useful and which showcase the rich functionality and flexibility that Visual Studio offers.

Getting Started

The best place to start with Visual Studio is to create a new project. This is the starting point for most development efforts and it lets me show you the different elements that make up Visual Studio.

Start Visual Studio and select `New Project` from the `File` menu. This will open the `New Project` window, as shown in Figure 1-1. Visual Studio can create a wide range of different projects types. For this chapter, I am using the Visual Studio 11 Express Beta release that is solely for Windows 8 Metro development, so all of the project types are Metro related. If you are using a different version of Visual Studio or have obtained it from MSDN rather than the Windows 8 preview website, then you will see a different set of available projects.



Type: Visual C#

A single-page project for a Windows Metro style app that has no predefined controls or layout.

Name: MetroSample

Location: C:\Projects

Solution name: MetroSample

□ Create directory for solution

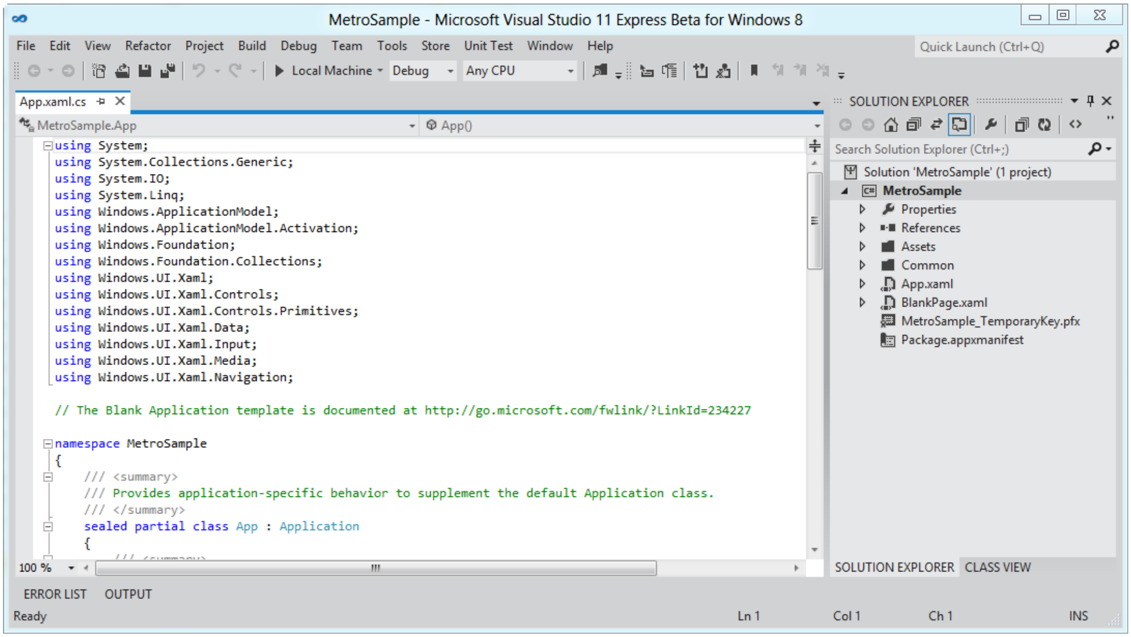
□ Add to source control

Figure 1-1. Creating a new Visual Studio project

The left panel lets you navigate through the project categories. To follow along with the examples in this chapter, you should navigate to `Installed > Templates > Visual C# > Windows Metro style`. As you do this, the middle panel will update to show you the individual project types in the category. I am going to use the `Blank Application` template for this chapter.

■ **Tip** For every project category, Microsoft defines a "blank" project template which contains just the bare essentials to get started and several "quick-start" templates which are populated for common scenarios. My advice is to stick with the blank project templates and add in the items you require as you need them. The quick-start templates are not well thought out and I find they just get in my way.

Once you have selected the project type, you use the text fields at the bottom of the window to enter the name of your project and the location where the project files will be stored. I have called my project `MetroSample` and told Visual Studio to save the project in the `C:\Projects` directory. Click `OK` to create the project. Visual Studio will take a moment to generate the files that you require and save them on the disk. Once the preparation has been completed, Visual Studio will open the project so that you can start development, as shown in Figure 1-2.



```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using Windows.ApplicationModel;
using Windows.ApplicationModel.Activation;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;

// The Blank Application template is documented at http://go.microsoft.com/fwlink/?LinkId=234227

namespace MetroSample
{
    /// <summary>
    /// Provides application-specific behavior to supplement the default Application class.
    /// </summary>
    sealed partial class App : Application
    {
```

Figure 1-2. *Opening a new project with Visual Studio*

The layout shown in Figure 1-2 is just the starting point and you can rearrange any part of the layout by dragging around. You can also detach any part of the layout from the main window and make it into a floating window, which is especially useful if you are using multiple monitors.

Understanding the Solution Explorer

The Solution Explorer allows you to navigate through the contents of your project. Most of the Solution Explorer content is related to the file structure of the project on the disk, but there are other items shown as well, including the assemblies that your project references. You can see a closer view of the Solution Explorer in Figure 1-3, where I have expanded some of the folders so you can see individual files.

■ **Tip** You might be wondering how you created a *project* but find yourself navigating through a *solution*. A Visual Studio solution is a container for one or more projects, allowing you to work on related activities in a joined up way. You can compile all of the projects in a solution together, for example, or create test projects which live alongside your main development project. Right-click on the top-most item in the `Solution Explorer` window and select the `Add > New Project` or `Add > Existing Project` menu items to add another project to the solution.

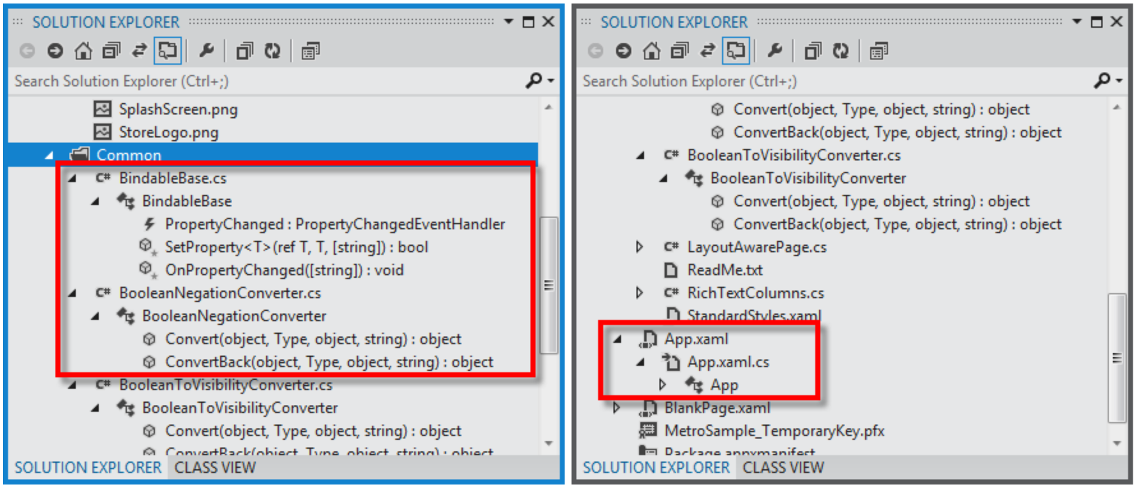
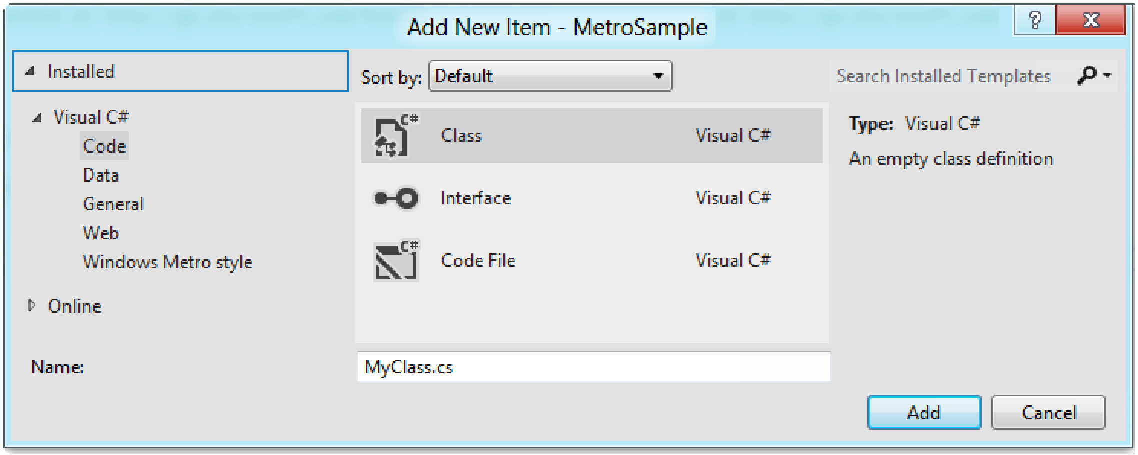

SOLUTION EXPLORER
Search Solution Explorer (Ctrl+;)
SplashScreen.png
StoreLogo.png
Common
BindableBase.cs
BindableBase
PropertyChanged : PropertyChangedEventHandler
SetProperty<T>(ref T, T, [string]) : bool
OnPropertyChanged([string]) : void
BooleanNegationConverter.cs
BooleanNegationConverter
Convert(object, Type, object, string) : object
ConvertBack(object, Type, object, string) : object
BooleanToVisibilityConverter.cs
BooleanToVisibilityConverter
Convert(object, Type, object, string) : object
SOLUTION EXPLORER
CLASS VIEW
SOLUTION EXPLORER
Search Solution Explorer (Ctrl+;)
Convert(object, Type, object, string) : object
ConvertBack(object, Type, object, string) : object
BooleanToVisibilityConverter.cs
BooleanToVisibilityConverter
Convert(object, Type, object, string) : object
ConvertBack(object, Type, object, string) : object
LayoutAwarePage.cs
ReadMe.txt
RichTextColumns.cs
StandardStyles.xaml
App.xaml
App.xaml.cs
App
BlankPage.xaml
MetroSample_TemporaryKey.pfx
SOLUTION EXPLORER
CLASS VIEW

Figure 1-3. *A closer view of the Solution Explorer*

You can use the Solution Explorer as a simple file browser, but it does a lot more. For example, you can expand the items for code files to see the classes they contain and the properties, events and methods that each class defines. For XAML files, the Solution Explorer associates the code-behind file as a child of the XAML item to keep the view clean and simple. For some file types, such as images, hovering the mouse over the item shows a preview of the file contents.

Adding New Items

You can manage, as well as view, the contents of your project using the Solution Explorer. Right-clicking on the Solution Explorer window brings up a menu from which you can add a project assembly, import content and resources and, if you wish, compile your code. When you are starting a new project, the most frequent task is to add new items to the project. To do this, right-click the folder you want the item to be created in (or select the project item for top-level items) and select `Add > New Item`. Visual Studio will display the `Add > New Item` window, as shown in Figure 1-4.



Add New Item - MetroSample

Sort by: Default

Type: Visual C#

An empty class definition

Name: MyClass.cs

Figure 1-4. *The Add New Item window*

The categories of items you can create are listed on the left and the individual item types are shown in the middle panel. (The right panel shows a preview of some types of items, but they are rarely illuminating). I am going to create a C# class file that I will use later in this chapter, which I do by selecting the `Class` item from the `Code` category. I give my new files a name (`MyClass.cs` in this case) and click the `Add` button to create the file. Visual Studio creates the file on the disk, adds the new item to the Solution Explorer and opens the file for editing.

■ **Tip** I tend to use the Solution Explorer to add items and references for my projects, but all of the same commands are available via the `Project` and `Debug` menus as well.

Three Other Solution Explorer Features

I don't have the space in this chapter to do into all of the Solution Explorer features in detail, so I am just going to give you some quick pointers to three other features worthy of further exploration.

First, you can search the entire solution using the search box at the top of the Solution Explorer window. I find this useful for quickly locating a method or filtering the view so that only one kind of file is visible (by searching for `.cs`, for example).

Second, you can focus on one particular part of a project by selecting an item, right-clicking and choosing `Scope to This` from the menu. The Solution Explorer will display only the item you have selected and its descendants, which is useful for working on small parts of large projects. Click the `Home` button at the top of the Solution Explorer window to switch back to showing all of the solution items.

Third, if you select an item, right-click and select `New > View`, the Solution Explorer will clone itself and narrow the scope of the new window to the selected item. I use this feature a lot because it lets me have multiple scoped views on large projects that would be unwieldy to navigate in a single window.

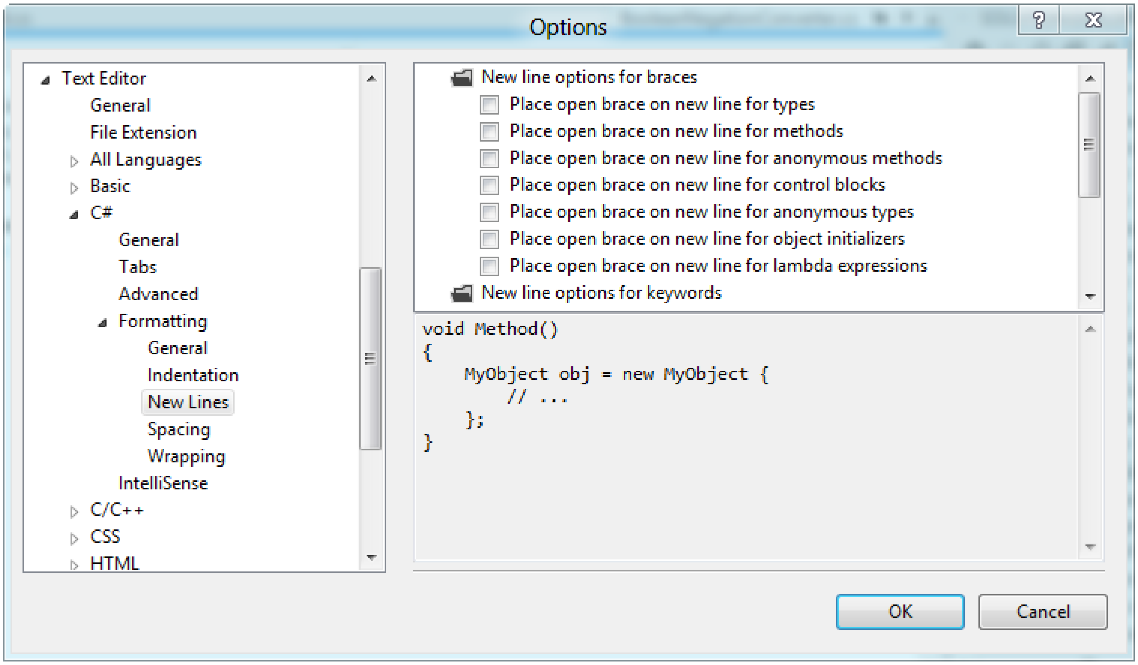
Understanding the Editor

The editor is the heart of any development environment and Visual Studio is no exception. In the sections that follow, I'll give you a quick tour around the editor and show you some of its most useful features.

Customizing the Editor

I like the Visual Studio editor, but only once I have configured it to suit my coding style. There are a lot of configuration options available and you can change most behaviors and formats that the editor applies. To change any Visual Studio setting, you select `Tools > Options`. The `Options` window that opens covers all of the Visual Studio configuration choices. Use the left panel to navigate to the `Text Editor` section. Here you will see groups of options that are applied to individual languages and to the editor as a whole.

I don't like the placement of C# braces that Visual Studio uses, so one of the first things I do is select the `C# > Formatting > New Lines` item on the left and work my way through the listed options. A preview of the effect of each option is shown at the bottom of the window, as Figure 1-5 illustrates.



New line options for braces

- ☐ Place open brace on new line for types
- ☐ Place open brace on new line for methods
- ☐ Place open brace on new line for anonymous methods
- ☐ Place open brace on new line for control blocks
- ☐ Place open brace on new line for anonymous types
- ☐ Place open brace on new line for object initializers
- ☐ Place open brace on new line for lambda expressions

New line options for keywords

```
void Method()
{
    MyObject obj = new MyObject {
        // ...
    };
}
```

Figure 1-5. *Changing the options for brace placement for C# files*

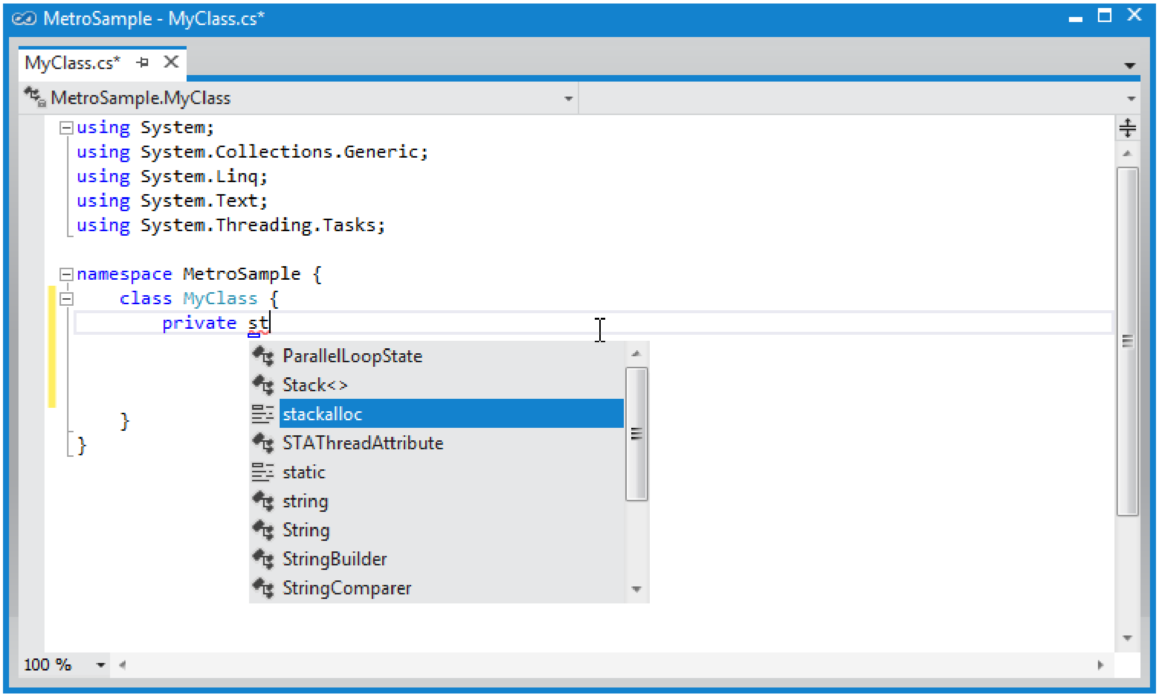
■ **Tip** The previews are useful but they don't take into account the choices you have made for other options, which means that it can take a few attempts at configuration before you find the combination that suits you.

Writing Code Using the Editor

At its heart, software development is still about grinding out the code and the Visual Studio editor has all of the features you would expect in a modern IDE to help make this is painless and error-free as possible. Keywords are color-coded and errors are underlined as you type.

On the left margin of the editor are collapsible regions which let you hide sections of code and right-clicking on any word shows a popup menu with options to refactor the code, jump to the original definition and to control which regions are displayed.

As you type, Visual Studio will display a popup window offering you suggestions - this includes types and members from classes within the project and the namespaces which have been brought into the scope of the code file with using statements, as shown in Figure 1-6.

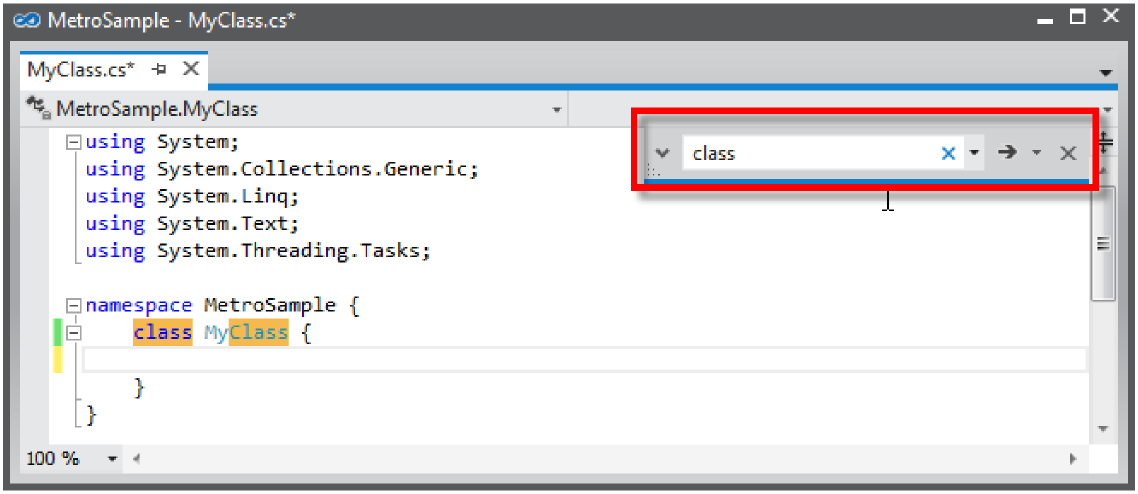


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MetroSample {
    class MyClass {
        private st


    }
}
```

Figure 1-6. Code completion in the Visual Studio code editor

Searching in Code

One of the editor features that I like the most in Visual Studio is the ability to easily and quickly search for text on a per-file basis. Type `Ctrl+F` or select `Edit > Find and Replace > Quick Find` and a small in-window search box appears, like the one shown in Figure 1-7. As you enter your search term, the editor highlights all the matching instances it can find and you can page through the matches to look at them in turn.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MetroSample {
    class MyClass {

    }
}
```

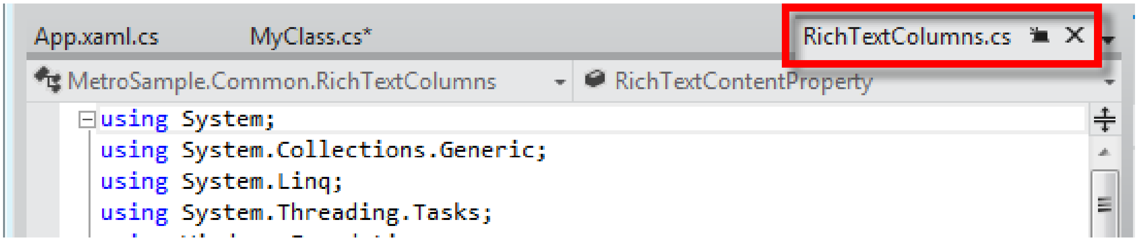
Figure 1-7. Using the quick find feature

In-file search is not a new idea, of course, but there is something pleasing about the way that this is implemented in Visual Studio, which makes searching quick and easy - and I like the fact that I can search within files independently of one another.

Tip Traditional search tools are available under the `Edit > Find and Replace` menu. See the `Find in Files` and `Replace in Files` menu items.

Peeking at Files

If you want to take a quick peek at a file then click it once in the Solution Explorer. The tab for the file will be shown at the right edge of the tab area with a special icon, as shown in Figure 1-8.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
```

Figure 1-8. Peeking at the contents of a file

Only one file at a time is displayed in this way, so if you click on a second file in the Solution Explorer, then it replaces the first one in the peek position. This allows you to quickly page through a set of files to locate something - I find myself using this feature most often to locate variable names and attribute values without being left with a dozen open file tabs.

The file will be opened if you edit the file or click on the icon in the tab and the tab itself will be moved to the left side of the tab group, as for regular files.

Grouping Files into Windows

There is a lot of flexibility in how you arrange your editor windows in Visual Studio. You can detach any editor tab from the main window by dragging it. The editor you drag is displayed in a new window and you can drag and release other tabs to join it. You can have as many windows as you need, which is useful on a multi-monitor configuration. You can dock Solution Explorer windows to floating code windows and essentially duplicate any combination of windows you like in any way you like.

■ **Tip** Hold Control and double-click on the title bar to return any detached window to the main Visual Studio layout.

You can create multiple editor windows for the same file by selecting `Window` > `New Window` and group files together by clicking on a file tab and selecting `New Vertical Tab Group` or `New Horizontal Tab Group` from the popup menu. Once you have created tab groups, you can allocate files to groups by dragging their tabs from group to group. I often use the vertical tab feature because it allows me to view several files at once on a large display.

Three Other Code Editor Features

There are a lot of helpful features in the code editor that I recommend you explore. In this section, I describe three of my favorites.

First, I mentioned earlier that I don't like the brace positioning that Visual Studio uses by default. I realize that this is a contentious topic and it can be painful to work on a team with split preferences. Visual Studio helps by making it easy to reformat a code file to your preferences. Select `Edit` > `Advanced` > `Format Document` to format the entire file or use the `Format Selection` option to transform just a portion.

Second, the Edit > Advanced menu also contains items to quickly comment and uncomment code blocks. The default keyboard shortcuts for this are not particularly useful, but you can change the key bindings by selecting Tools > Customize and clicking the Keyboard button. The drawback with the automatic commenting is that it doesn't detect when a code block is already commented out and just adds another set of comment characters.

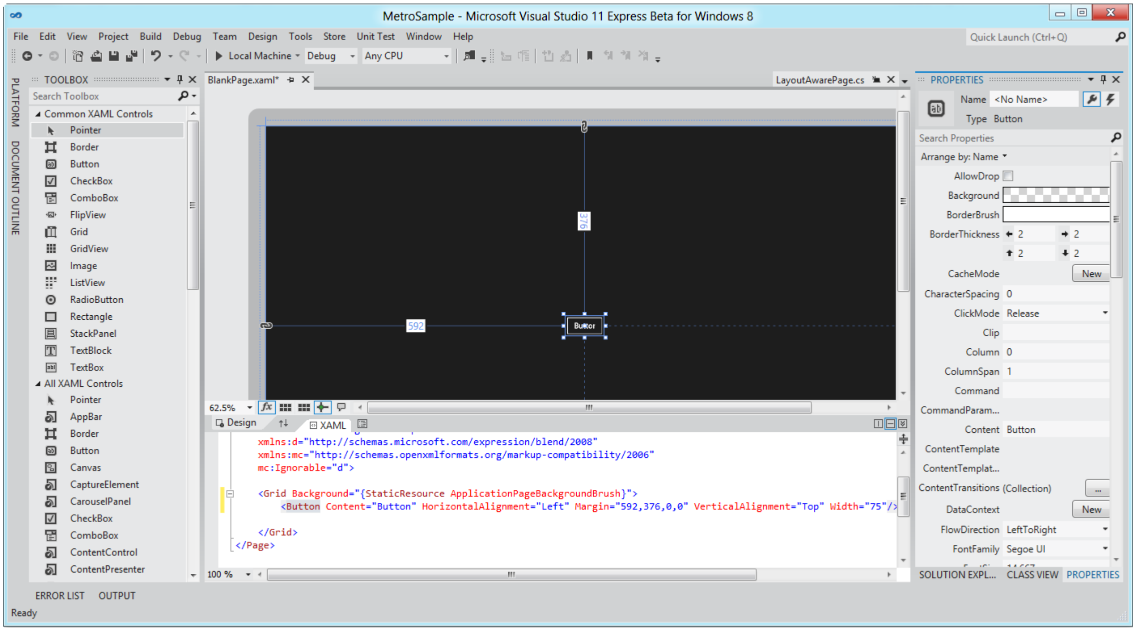
Third, you can see all the calls to and from a member by selecting it in the editor, right-clicking and picking View Call Hierarchy from the menu. This is not a feature which is unique to Visual Studio, but it is well implemented, fast and easy to use.

Understanding the Design Surface

For some file types, Visual Studio offers an alternative to text editing. The *design surface* is a drag and drop visual UI editor which can be used with XAML and ASP.NET Web Forms. To see the design surface in action, double click on BlankPage.xaml file in the Solution Explorer.

■ **Tip** The design surface can be useful, but it has many flaws. I find it useful when I am building the rough layout for an application at the start of a project, but I find it unreliable for complex projects. It crashes often, although this doesn't interrupt Visual Studio and you can simply reload the content to give it another chance. The design surface is one of the weakest parts of Visual Studio.

You may need to open another couple of windows to get the most from the design surface, both of which can be found under the View menu. The Toolbox window lists the UI and data controls that you can drag to the design surface. The Properties Window allows you to configure the contents of the design surface by assigning new values to properties. You can see the default arrangement of these windows alongside the design surface in Figure 1-9.



```
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d">

    <Grid Background="{StaticResource ApplicationPageBackgroundBrush}">
        <Button Content="Button" HorizontalAlignment="Left" Margin="592,376,0,0" VerticalAlignment="Top" Width="75"/>

    </Grid>
</Page>
```

Name <No Name>
Type Button
AllowDrop □
BorderThickness 2 2 2 2
CharacterSpacing 0
ClickMode Release
Column 0
ColumnSpan 1
Content Button
ContentTransitions (Collection)
FlowDirection LeftToRight
FontFamily Segoe UI

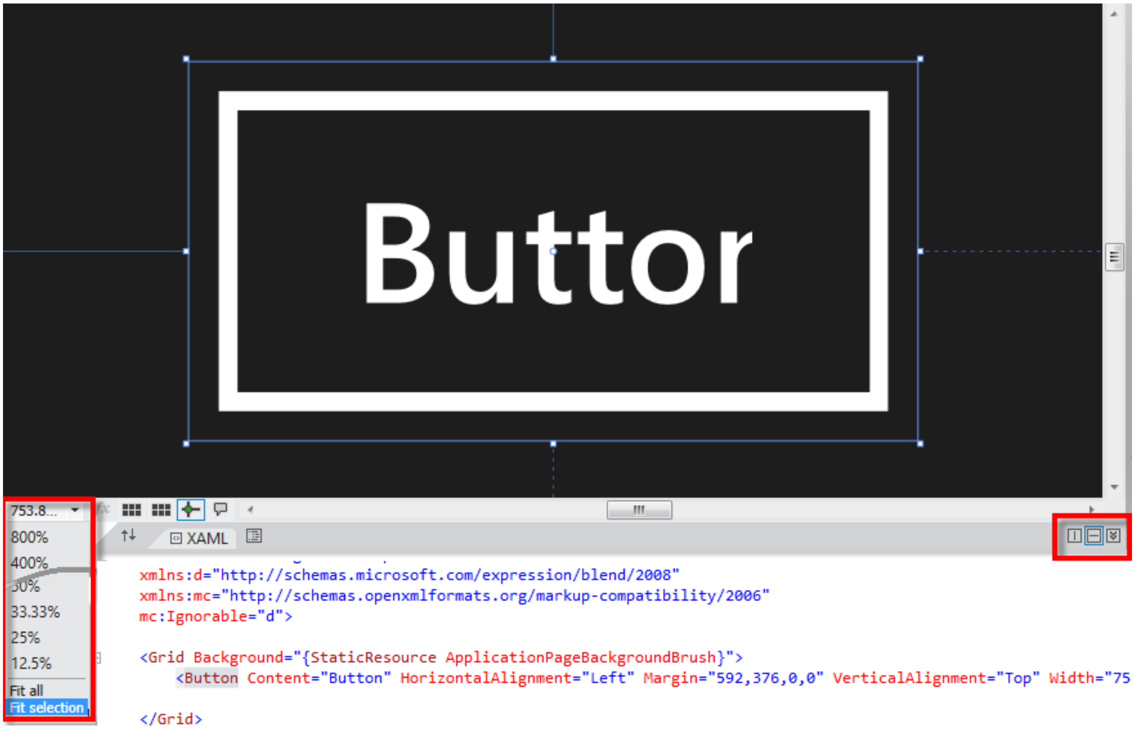
Figure 1-9. Using the design surface

The largest window in the layout is the design surface. Beneath the design surface is a text editor containing the XAML code. In a Metro app, the design surface for a XAML file is made to look like a tablet with borders. I created the display in Figure 1-9 on the design surface by dragging the `Button` item from the `Toolbox` window and dropping it on the surface.

Scaling and Positioning Content

It can be a little hard to make out the details when the entire layout is being displayed on the design surface, so the first thing I am going to show you is how to zoom the content. At the bottom of the design surface, just above the XAML editor you will see a small row of commands - so small that you can easily miss them.

Clicking on the down arrow on the left-most command opens a drop-down menu that you can use to change the design surface for zoom factor. Figure 1-10 shows the effect of the `Fit Selection` option, which magnifies the currently selected item so that it fills the design surface window. There are less extreme zoom options available as well, but for getting down into the detail, the `Fit Selection` option is ideal.



```
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
mc:Ignorable="d">

<Grid Background="{StaticResource ApplicationPageBackgroundBrush}">
    <Button Content="Button" HorizontalAlignment="Left" Margin="592,376,0,0" VerticalAlignment="Top" Width="75

</Grid>
```

__Figure 1-10.__ Changing the zoom factor for the design surface

In Figure 1-10, I have highlighted controls on the far-right of the divider bar. These let you change the layout, so that the XAML and design surface are split vertically or the XAML is hidden entirely. It can take a while to find a combination that suits your preferences and works with your screen space, but it is worth experimenting until you get something you like.

Editing Items on the Design Surface

You configure the items on the design surface by changing the values shown in the `Properties` window. As an example, you can see in Figure 1-10 that the button I added to the design surface doesn't have enough space to display its text content properly and that the content itself is pretty useless.

To change the text displayed by the button, I expand the `Common` property category and assign a new value to the `Content` property. In this example, I will change the text to `Press Me`. The design surface updates as soon as the focus moves away from the text field for the `Content` property, giving me immediate feedback. To change the size of the text, I expand the `Text` category and enter a new font size. As you can see in Figure 1-11, some properties are grouped together. For the `Text` category, the properties are arranged in tabbed groups, much like you would see in a word processor.

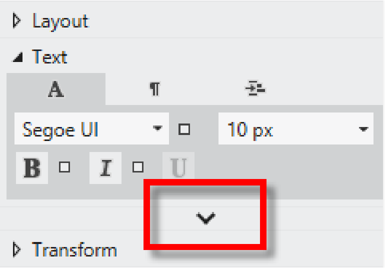


Figure 1-11. *Working with grouped properties in the Properties window*

Visual Studio goes to some effort to focus on the most commonly-used properties for a control. I have highlighted the down arrow in the figure, which is one of the ways that this focus is maintained. The text-related properties you are most likely to want are displayed as soon as you open the Text category, but you can see and edit some less common properties by clicking the arrow.

When you change the value for a property, the XAML for the control is updated automatically. For my example changes, the XAML for the button element has been updated with new values for the Content and FontSize properties. You can see these changes in Listing 1-1, which shows the BlankPage.xaml file.

Listing 1-1. *Property updates reflected in XAML*

```
<Page
    x:Class="MetroSample.BlankPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:MetroSample"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d">

    <Grid Background="{StaticResource ApplicationPageBackgroundBrush}">
        <Button Content="Press Me" HorizontalAlignment="Left" Margin="592,376,0,0"
            VerticalAlignment="Top" Width="75" FontSize="10"/>
    </Grid>
</Page>
```

Creating Event Handlers

The Properties window helps you set up event handlers for your UI controls. To switch to the events view, click on the icon at the top of the Properties window which looks like a bolt of lightning, as shown in Figure 1-12.

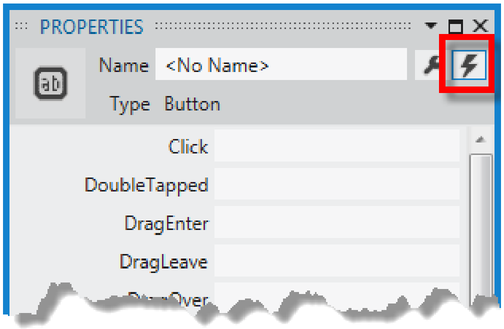


Figure 1-12. The Properties window event view

The Properties window switches to a list of events that the selected UI control supports. For a button, the most useful is the `Click` event, which is the first one shown in the list. Double-click in the empty text field for the `Click` event and Visual Studio will create a handler method in the code-behind C# file, complete with the appropriate parameters. Visual Studio will also switch to the code editor so you can start to add the code you require, as shown in Listing 1-2.

Listing 1-2. An event handler generated by Visual Studio

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace MetroSample {
    /// <summary>
    /// An empty page that can be used on its own or navigated to within a Frame.
    /// </summary>
    public sealed partial class BlankPage : Page {
        public BlankPage() {
            this.InitializeComponent();
        }

        /// <summary>
        /// Invoked when this page is about to be displayed in a Frame.
        /// </summary>
        /// <param name="e">Event data that describes how this page was reached.
        ///  The Parameter property is typically used to configure the page.</param>
        protected override void OnNavigatedTo(NavigationEventArgs e) {
        }

        private void Button_Click_1(object sender, RoutedEventArgs e) {

        }
    }
}
```

■ **Tip** You can also pick from methods which have the right signature to handle the event if they are already defined in the code-behind class. This requires the C# code to be compiled before the methods can be selected - I'll show you to compile a project shortly.

The XAML for the control is automatically updated to register the new event handler method, as shown in Listing 1-3.

Listing 1-3. *Automatically updated XAML reflecting an event handler method*

```
...
<Button Content="Press Me" HorizontalAlignment="Left" Margin="592,376,0,0"
    VerticalAlignment="Top" Width="75" FontSize="10" Click="Button_Click_1"/>
...
```

Three Other Design Surface Features

In keeping with the other sections of this chapter, there are three features of the design surface that you should take the time to explore. The first feature, and the one that I rely on most, is that the relationship between the XAML text and the design surface works in both directions. This means that you can edit the XAML directly and see the effect in the layout immediately. I find this useful when tracking down odd layout problems and I find working directly with XAML to be faster and easier than working from the design surface directly.

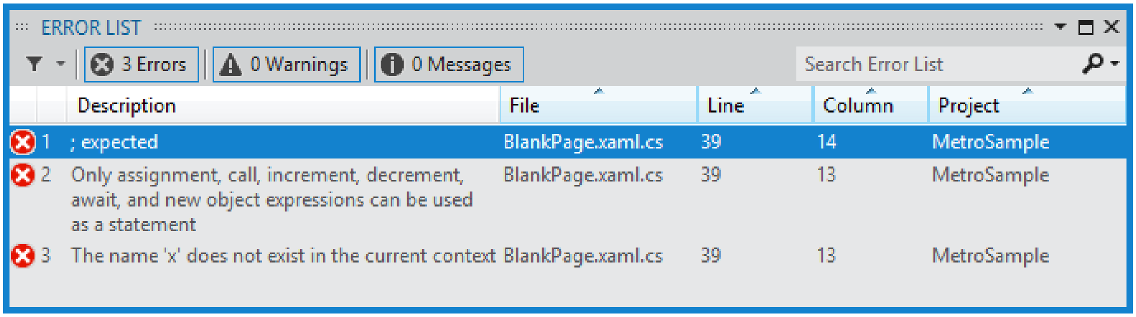
The second feature is the small button with an up and down arrow near to the design surface zoom setting. Clicking this button reverse the position of the XAML editor and the design surface. Combine this with the vertical and horizontal split buttons and you can create an arrangement of design and code tools which suits your personal working style.

The final feature is the search box at the top of the list of properties in the Properties Window. Start typing in this box and only the properties whose names match the string you have typed will be displayed. Some XAML controls have a lot of properties and they are not always organized into categories in a logical way, so being able to find a property quickly can save you several minutes of digging around for a particular property.

Compiling and Debugging

The commands for compiling the project can be found in the Build menu. The Build Solution menu item does an incremental build, such that Visual Studio will try to only compile files which have been modified since the last compile. The Rebuild Solution compiles everything from scratch, including those items that have not changed.

The Error List window is used to display any errors that are reported during the build process. To create an error, I just typed the letter x as the body for the event handler method created in the last section. You can see the entries this causes in the Error List window in Figure 1-13.



	Description	File	Line	Column	Project
1	; expected	BlankPage.xaml.cs	39	14	MetroSample
2	Only assignment, call, increment, decrement, await, and new object expressions can be used as a statement	BlankPage.xaml.cs	39	13	MetroSample
3	The name 'x' does not exist in the current context	BlankPage.xaml.cs	39	13	MetroSample

Figure 1-13. Compilation errors in the Errors List window

Visual Studio constantly compiles code files as you edit them and so you can often see simple errors highlighted in the text before you perform a build. Not all errors are detected in this way and so you'll often see more serious problems for the first time when you compile. Each error is described alongside details of where in the source code you need to focus and double-clicking on an error will open the file that contains the problem and jump to the right line automatically.

Starting the Debugger

Compile-time errors are only part of the story - you will often need to track down behavior problems that occur when your project is running. For this, you will need the debugger. Visual Studio has one of the best debuggers that I have used. It is quick, fully featured and easy to use.

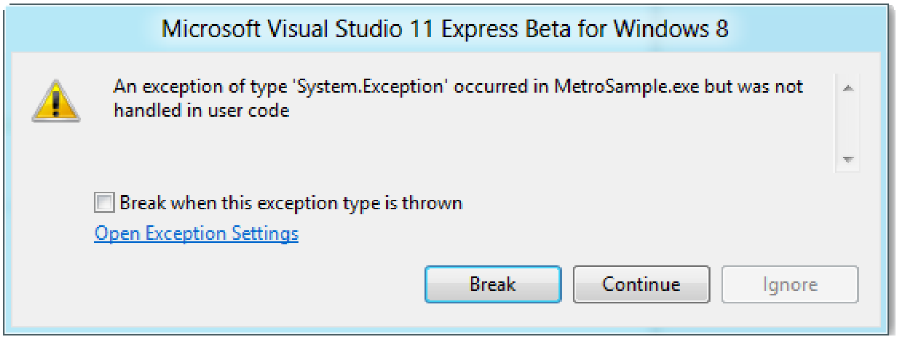
I want to give the debugger something to work with, so I have added a statement to the event handler method in the BlankPage.xaml.cs file that throws an exception, as shown in Listing 1-4.

Listing 1-4. Throwing an exception

```
...
private void Button_Click_1(object sender, RoutedEventArgs e) {
    throw new Exception();
}
...
```

To start your app with the debugger, select Start Debugging from the Debug menu. Since my example is a Metro app project, you will see the app start and the sole UI control, the button, being displayed. Clicking the button causes the statement I added in Listing 1-4 to be executed and my exception to be thrown. This exception isn't handled anywhere else in the code, and so it will make the debugger pause the app and present me with the choice to break or continue execution, as shown in Figure 1-14.

■ **Tip** Select `Start Without Debugging` from the `Debug` menu if you want to see how your app runs without the debugger.



Microsoft Visual Studio 11 Express Beta for Windows 8

An exception of type 'System.Exception' occurred in MetroSample.exe but was not handled in user code

□ Break when this exception type is thrown

Figure 1-14. *Breaking or continuing execution of the project*

If you click `Continue`, the debugger will resume execution of the app and try to carry on as though nothing had gone wrong. If you click `Break`, the debugger will pass control of the app to you so that you can investigate the problem. The current statement is shown in yellow and you can control the execution with the `Step Into`, `Step Over` or `Step Out` items from the `Debug` menu. (There are also icons in the debugging toolbar which perform the same functions).

You can open windows which will display the current call stack, the set of local variables and create watch lists - all of the features you would expect from a modern debugger. There are some nice additional touches - for example, if you hover the mouse over a variable, a popup window will display its value. If the variable is assigned a complex object, then you can expand the popup to reveal details of its inner structure and state. You can pin the popup to the window and it will continue to be displayed and updated as the object it is tracking is updated.

You can resume execution of the app by selecting `Continue` from the Debug menu or halt execution by selecting `Stop Debugging`.

Setting Breakpoints

You can create a breakpoint if you want the debugger to halt execution at a particular point in the execution of your app, even if no exception has been encountered. Right-click the statement you want to break at and select `Breakpoint` > `Insert Breakpoint` from the popup menu. A red circle will be displayed at the left margin of the text editor. You can define multiple breakpoints in the same file or throughout a project and whenever the debugger reaches one, execution will be halted and control passed to you. Select `Continue` when you are done and the app will run until the next breakpoint is encountered (or, if the app is performing a loop, the same breakpoint comes around again). You can remove or disable breakpoints by right-clicking on the red dot in the margin or by selecting `Delete All Breakpoints` or `Disable All Breakpoints` from the `Debug` menu.

Three Other Build Features

In keeping with the other sections of this chapter, there are three build features that you should take the time to explore. The first two are both features of the Error List window that allow you to focus on the errors that matter. First, there is a search box at the top-right of the window. If you start typing, the set of errors will be filtered to those which contain the string you have entered. Like most searches in Visual Studio, the results are generated each time you type a key, giving you immediate feedback.

The second feature is the filter button at the top-left of the Error List window - this is another button that is so small that it is easy to miss it. Clicking on this button lets you filter the errors so that errors are only shown if they are in the files which you have open for editing, from the current project selected in the Solution Explorer or from the current document.

The final feature I want to mention is the Visual Studio support for performance analysis. If you select `Start Performance Analysis` from the `Debug` menu, Visual Studio will run your project and collect detailed performance information. When the app exits, the data is crunched and you can explore the performance characteristics of you app. It can take a while to get useful data out of the performance analysis, but it is worth the investment.

Summary

In this chapter, I have given you a whistle-stop tour of Visual Studio, focusing on the major building blocks that you will use to create, edit, build and test your projects. Visual Studio is a rich and complex environment. It is worth taking the time to explore beyond the basics, to configure the environment to your workflow and preferences and to learn how to get the best out of the excellent testing and debugging tools.

Chapter 2 introduces the basic of Metro app development a little more specifically. For further information about the topics introduced in this chapter and more on Visual Studio generally, please see my *Pro Visual Studio 11* book, due out this fall from Apress. For more in-depth coverage of Windows 8 development, please see *Pro Windows 8 Development with XAML and C#*, by Jesse Liberty (Apress 2012).

CHAPTER 2

Getting Started with Metro

Adapted from Chapter 1 of *Metro Revealed: Building Windows 8 apps with HTML5 and JavaScript*

Metro apps are an important addition to Microsoft Windows 8, providing the cornerstone for a single, consistent programming and interaction model across desktops, tablets, and smartphones. The Metro app user experience is very different from previous generations of Windows applications: Metro-style apps are full-screen and favor a usability style that is simple, direct, and free from distractions.

Metro apps represent a complete departure from previous versions of Windows. There are entirely new APIs, new interaction controls, and a very different approach to managing the life cycle of applications.

Metro apps can be developed using a range of languages, including JavaScript, Visual Basic, C++, and, the topic of this book, C#. Windows 8 builds on the familiar to let developers use their existing C# and XAML experience to build rich Metro apps and integrate into the wider Windows platform

■ **Note** Microsoft uses the terms *Metro style* and *Metro-style app*. I can't bring myself to use these awkward terms, so I am just going to refer to *Metro* and *Metro apps*. I'll leave you to mentally insert *style* as needed.

If you have worked on WPF or Silverlight projects, then you will have enough XAML knowledge to build Metro apps. Don't worry if you haven't worked with XAML before; you'll can pick it up as you go, and I give you a brief overview later in this chapter to get you started.

What Software Do You Need for This Book?

You will need the *Windows 8 Consumer Preview* and the *Visual Studio 11 Express Beta for Windows 8*. You can download both of these from `http://preview.windows.com`. You don't need any other tools to develop applications or for the examples in this book.

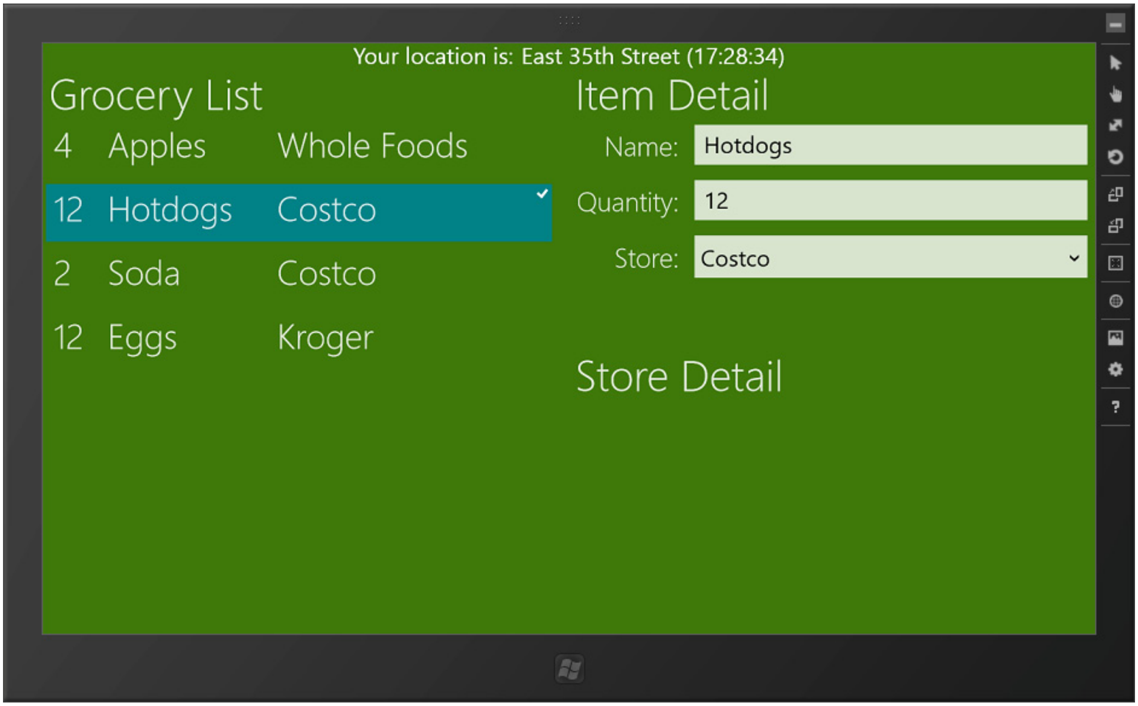
Windows 8 Consumer Preview is not a finished product, and it has some stability issues. You'll get the best experience if you install Windows 8 directly onto a well-specified PC, but you can get by with a virtual machine if you are not ready to make the switch.

If you do want more comprehensive coverage of Metro development, then Apress will be publishing Jesse Liberty's *Pro Windows 8 Development with XAML and C#* book for the final release of Windows 8. They will also be publishing my *Pro Windows 8 Development with HTML5 and JavaScript* if you want to use more web-oriented technologies to build your Metro apps.

Getting Started

In this chapter I will give you a brief overview of XAML, take you on a tour of the important files in a Metro development project, show you how to run your Metro apps in the Visual Studio simulator, and explain how to use the debugger.

The example application for this chapter is a simple grocery list manager called *MetroGrocer*. While we will not get very far with this example in this chapter, if you procure the complete book, *Metro Revealed: Building Windows 8 apps with XAML and C#* you will learn how the full application was built. In Figure 2-1, you can see how the app looks.



Your location is: East 35th Street (17:28:34)

Grocery List

4	Apples	Whole Foods
12	Hotdogs	Costco ✓
2	Soda	Costco
12	Eggs	Kroger

Item Detail

Name: Hotdogs

Quantity: 12

Store: Costco

Store Detail

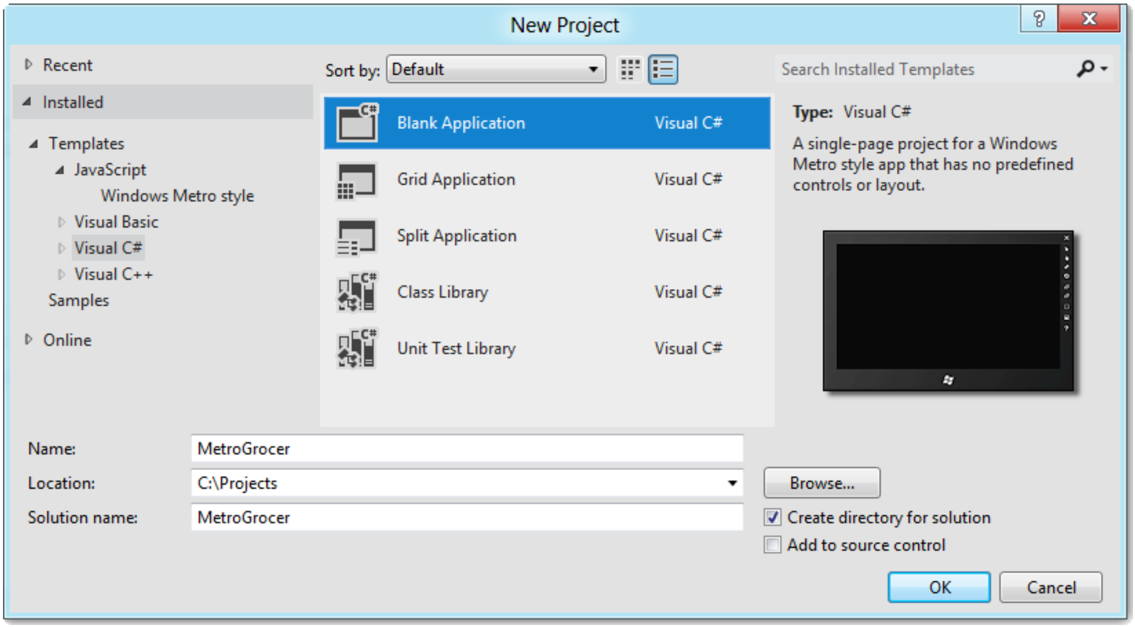
Figure 2-1. The example application

Getting Up and Running

In this section, I'll create the project for the example application and show you each of the project elements that Visual Studio generates. I'll break this process down step by step so that you can follow along. This will all be familiar if you read the preceding chapter. If you prefer, you can download the ready-made project from Apress.com.

Creating the Project

To create the example project, start Visual Studio and select File, New Project. In the New Project dialog, select Visual C# from the Templates section on the left of the screen, and select Blank Application from the available project templates, as shown in Figure 2-2.



Sort by: Default

Blank Application	Visual C#
Grid Application	Visual C#
Split Application	Visual C#
Class Library	Visual C#
Unit Test Library	Visual C#

Type: Visual C#

A single-page project for a Windows Metro style app that has no predefined controls or layout.

Name: MetroGrocer

Location: C:\Projects

Solution name: MetroGrocer

☑ Create directory for solution

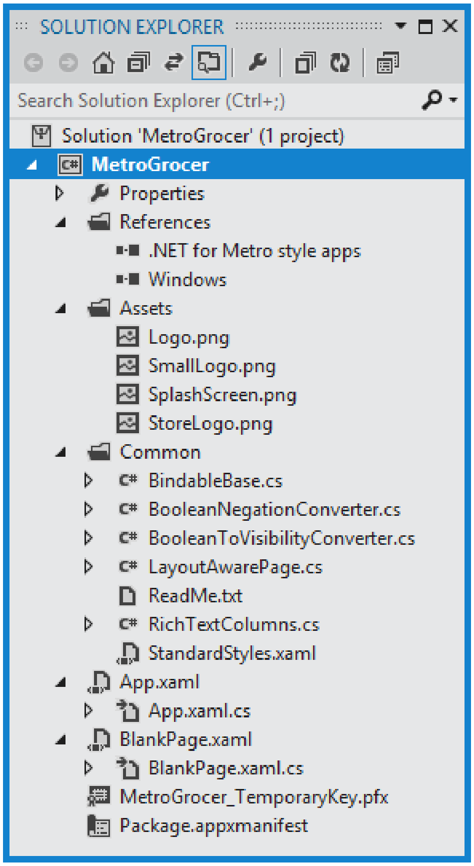
□ Add to source control

Figure 2-2. *Creating the example project*

Set the name of the project to MetroGrocer, and click the OK button to create the project. Visual Studio will create and populate the project.

Figure 2-3 shows the contents of the new project as displayed by the Visual Studio Solution Explorer. In the sections that follow, I'll describe the most important files in the project.

■ **Tip** Don't worry if the purpose or content of these files isn't immediately obvious. At this stage, I just want you to get a feel for how a Visual Studio Metro project fits together and what the important files are.



Solution 'MetroGrocer' (1 project)
MetroGrocer
Properties
References
.NET for Metro style apps
Windows
Assets
Logo.png
SmallLogo.png
SplashScreen.png
StoreLogo.png
Common
BindableBase.cs
BooleanNegationConverter.cs
BooleanToVisibilityConverter.cs
LayoutAwarePage.cs
ReadMe.txt
RichTextColumns.cs
StandardStyles.xaml
App.xaml
App.xaml.cs
BlankPage.xaml
BlankPage.xaml.cs
MetroGrocer_TemporaryKey.pfx
Package.appxmanifest

Figure 2-3. *The contents of a Visual Studio project created using the Blank Application template*

■ **Tip** Metro apps use a slimmed-down version of the .NET Framework library. You can see which namespaces are available by double-clicking the .Net for Metro style apps item in the References section of the Solution Explorer.

Exploring the App.xaml File

The `App.xaml` file and its code-behind file, `App.xaml.cs`, are used to start the Metro app. The main use for the XAML file is to associate `StandardStyles.xaml` from the `Common` folder with the app, as shown in Listing 2-1.

Listing 2-1. *The App.xaml File*

```
<Application
    x:Class="MetroGrocer.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:MetroGrocer">

    <Application.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source="Common/StandardStyles.xaml"/>
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Application.Resources>
</Application>
```

I'll discuss the `StandardStyles.xaml` file shortly, and, later in this chapter, I'll update `App.xaml` to reference my own resource dictionary. The code-behind file is much more interesting and is shown in Listing 2-2.

Listing 2-2. *The App.xaml.cs File*

```
using Windows.ApplicationModel;
using Windows.ApplicationModel.Activation;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;

namespace MetroGrocer {
    sealed partial class App : Application {

        public App() {
            this.InitializeComponent();
            this.Suspending += OnSuspending;
        }

        protected override void OnLaunched(LaunchActivatedEventArgs args) {
            if (args.PreviousExecutionState == ApplicationExecutionState.Terminated) {
                //TODO: Load state from previously suspended application
            }

            // Create a Frame to act navigation context and navigate to the first page
            var rootFrame = new Frame();
            rootFrame.Navigate(typeof(BlankPage));

            // Place the frame in the current Window and ensure that it is active
            Window.Current.Content = rootFrame;
            Window.Current.Activate();
        }

        void OnSuspending(object sender, SuspendingEventArgs e) {
            //TODO: Save application state and stop any background activity
        }
    }
}
```

Metro apps have a very specific life-cycle model, which is communicated via the `App.xaml.cs` file. It is essential to understand and embrace this model. That is beyond the scope of this chapter. For the moment, you need to know only that the `OnLaunched` method is called when the app is started and that a new instance of the `BlankPage` class is loaded and used as the main interface for the app.

■ **Tip** For brevity, I have removed most of the comments from these files and removed the namespace references that are not used by the code in the class.

Exploring the BlankPage.xaml File

Pages are the basic building blocks for a Metro app. When you create a project using the Blank Application template, Visual Studio creates a blank page, which it unhelpfully names `BlankPage.xaml`. Listing 2-3 shows the content of the `BlankPage.xaml` file, which contains just enough XAML to display…well, a blank page.

Listing 2-3. *The Contents of the BlankPage.xaml File*

```
<Page
    x:Class="MetroGrocer.BlankPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:MetroGrocer"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d">

    <Grid Background="{StaticResource ApplicationPageBackgroundBrush}">
    </Grid>
</Page>
```

If you have used XAML before, you will recognize the `Grid` control. Metro UI controls work in generally the same way as those from WPF or Silverlight, but there are fewer of them, and some of the advanced layout and data binding features are not available. The code-behind file for `BlankPage.xaml` will also be familiar if you have XAML experience, as shown in Listing 2-4.

■ **Tip** Don't worry about the XAML and code-behind files for the moment; I provide a quick overview later in this chapter.

Listing 2-4. *The Contents of the BlankPage.xaml.cs File*

```
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Navigation;

namespace MetroGrocer {

    public sealed partial class BlankPage : Page {
        public BlankPage() {
            this.InitializeComponent();
        }

        protected override void OnNavigatedTo(NavigationEventArgs e) {
        }
    }
}
```

Exploring the StandardStyles.xaml File

The `Common` folder contains files that are used by Visual Studio project templates. The only file I care about in this folder is `StandardStyles.xaml`, which is the resource dictionary file referred to in the `App.xaml` file (as shown in Listing 2-1). The `StandardStyles.xaml` file contains some of the styles and templates that make it easier to create an app that has an appearance that is consistent with the broader Metro look and feel. I am not going to list the complete file because it contains a lot of content, but Listing 2-5 shows an example of a text-related style.

Listing 2-5. *A Style from the StandardStyles.xaml File*

```
...
<Style x:Key="HeaderTextStyle" TargetType="TextBlock"
        BasedOn="{StaticResource BaselineTextStyle}">
    <Setter Property="FontSize" Value="56"/>
    <Setter Property="FontWeight" Value="Light"/>
    <Setter Property="LineHeight" Value="40"/>
    <Setter Property="RenderTransform">
        <Setter.Value>
            <TranslateTransform X="-2" Y="8"/>
        </Setter.Value>
    </Setter>
</Style>
...
```

■ **Caution** Don't edit the files in the `Common` folder. Rather, you should create and reference a custom resource dictionary.

Exploring the Package.appxmanifest File

The final file worth mentioning is the application manifest, called `Package.appxmanifest`. This is an XML file that provides information about your Metro app to the Windows runtime. You can edit this file as raw XML, but Visual Studio provides a nice properties-based editor to use instead.

An Incredibly Brief XAML Overview

Don't worry if you haven't used XAML before. The learning curve for creating Metro apps will be steeper, but you have the advantage of not expecting features from other XAML application types that are not available in Metro.

At its heart, XAML creates user interfaces declaratively, rather than in code. So, if I wanted to add a couple of button controls to my project, I add some markup to my XAML file, as shown in Listing 2-6.

Listing 2-6. *Adding Controls to the XAML Document*

```
<Page
    x:Class="MetroGrocer.BlankPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:MetroGrocer"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d">

    <Grid Background="{StaticResource ApplicationPageBackgroundBrush}">
        <StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">
            <Button x:Name="FirstButton" HorizontalAlignment="Center"
                    Click="ButtonClick">Click Me!</Button>
            <Button Style="{StaticResource TextButtonStyle}"
                    HorizontalAlignment="Center"
                    Click="ButtonClick">Or Click Me!</Button>
        </StackPanel>
    </Grid>
</Page>
```

The tag name in a XAML element specifies the control that will be added to the layout. I have added one `StackPanel` and two `Button` controls to my project. The `StackPanel` is a simple container that helps add structure to the layout; it positions its child controls in either a horizontal or vertical line (a *stack*). The `Button` controls are just what you'd expect: a button that emits an event when the user clicks it.

The hierarchical nature of the XML is translated into the hierarchy of UI controls. By placing the `Button` elements inside the `StackPanel`, I have specified that the `StackPanel` is responsible for the layout of the `Button` elements.

Using the Visual Studio Design Surface

You can do everything in C# in a Metro project and not use XAML at all. But there some compelling reasons to adopt XAML. The main advantage is that the design support for XAML in Visual Studio is pretty good and will, for the most part, show you the effect of changes to your XAML files in real time. As Figure 2-4 shows, Visual Studio reflected the addition of the `StackPanel` and `Button` elements on its XAML design surface, which I introduced briefly in Chapter 1. This isn't the same as running the application, but it is a broadly faithful representation; this isn't available for interfaces created in C#.

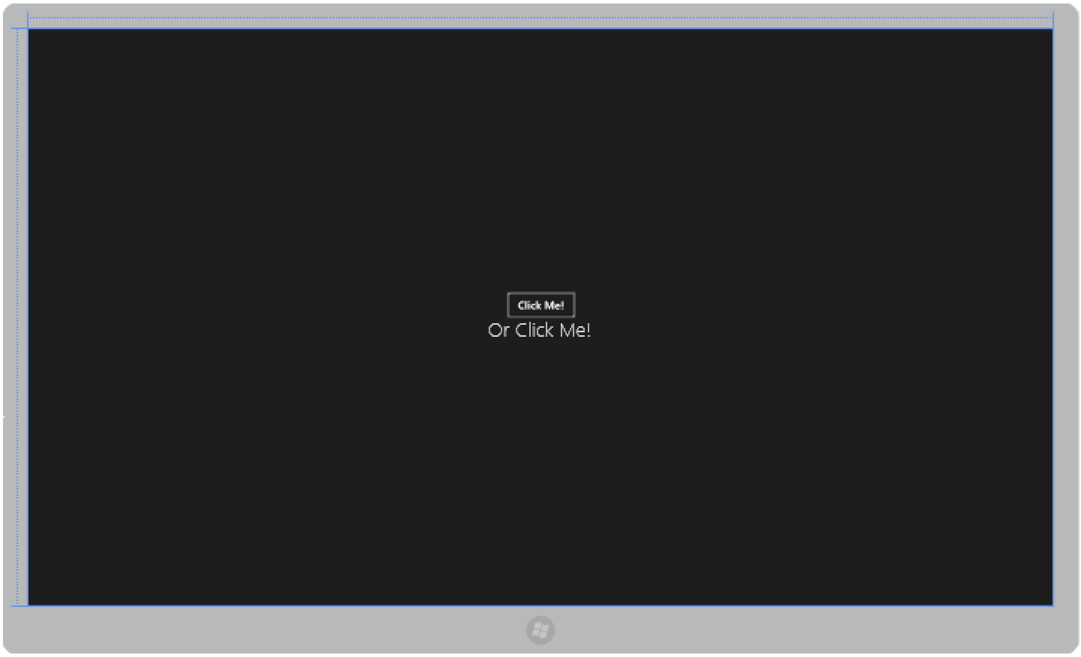


Figure 2-4. *Visual Studio reflecting the contents of a XAML file on its design surface*

Although XAML tends to be verbose, Visual Studio does a lot to make creating and editing it easier; it has some excellent autocomplete features that offer suggestions for tag names, attributes, and values. You can also design an interface by dragging controls from the Toolbox directly onto the design surface and configuring them using the Properties window. If neither of those approaches suits you, there is support for creating the XAML for Metro apps in the beta of Blend for Visual Studio, which was installed on your machine as part of the Visual Studio setup. I favor writing the XAML directly in the code editor, but then I am crusty old-school programmer who has never really trusted visual design tools, even though they have become pretty good in recent years. You may not be quite as crusty, and you should try the different styles of UI development to see which suits you. For the purposes of this book, I'll show you changes to the XAML directly.

Configuring Controls in XAML

You configure Metro UI controls by setting attributes on the corresponding XAML element. So, for example, I want the StackPanel to center its child controls. To do this, I set values for the HorizontalAlignment and VerticalAlignment attributes, like this:

```
...
<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">
...
```

Applying Styles

Attributes are also used to apply styles to UI controls. As an example, I applied the TextButtonStyle that is defined by Microsoft in the StandardStyles.xaml file:

```
...
<Button Style="{StaticResource TextButtonStyle}" HorizontalAlignment="Center"
    Click="ButtonClick">Or Click Me!</Button>
...
```

There are different ways to define and reference styles in XAML. I have used StaticResource to specify the style I want, but there are options for getting the style information from all sorts of sources. I am going to keep things simple in this chapter and stick to the basics wherever possible, focusing on the features that are specific to Metro apps.

Specifying Event Handlers

To specify a handler method for an event, you simply use the element attribute that corresponds to the event you require, like this:

```
...
<Button x:Name="FirstButton" HorizontalAlignment="Center"
    Click="ButtonClick">Click Me!</Button>
...
```

I have specified that the click event (which is triggered when the user clicks the button) will be handled by the ButtonClick method. Visual Studio will offer to create an event handler method for you when you apply an event attribute; I'll show you the other side of this relationship in in the next section.

Configuring Controls in Code

It relies on some clever compiler tricks and a C# feature known as *partial classes*. The markup in a XAML file is converted and combined with the code-behind file to create a single .NET class. This can seem a bit odd at first, but it does allow for a nice hybrid model where you can define and configure controls in XAML, the code-behind C# class, or both.

The simplest way of demonstrating this relationship is to show you the implementation of the event handler that I specified for the Button elements in the XAML file. Listing 2-7 shows the BlankPage.xaml.cs file, which is the code-behind file for BlankPage.xaml.

Listing 2-7. *The BlankPage.xaml.cs File*

```
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Navigation;

namespace MetroGrocer {

    public sealed partial class BlankPage : Page {
        public BlankPage() {
            this.InitializeComponent();
        }

        protected override void OnNavigatedTo(NavigationEventArgs e) {
        }

        private void ButtonClick(object sender, RoutedEventArgs e) {
            System.Diagnostics.Debug.WriteLine("Button Clicked: "
                + ((Button)e.OriginalSource).Content);
        }
    }
}
```

I am able to refer to the `ButtonClick` method without any qualification in the XAML file because the code generated from the XAML file is merged with the C# code in the code-behind file to create a single class. The result is that when one of the `Button` elements in the app layout is clicked, my C# `ButtonClick` method is invoked.

■ **Tip** No console is available for Metro apps, so use the static `System.Diagnostcs.Debug.WriteLine` method if you want to write out messages to help figure out what's going on in your app. These are shown in the Visual Studio Output window, but only if you start your app using Start With Debugging from the Debug menu.

This relationship goes both ways. Notice that some of the elements in the XAML file have a `x:Name` attribute, like this:

```
...
<Button x:Name="FirstButton" HorizontalAlignment="Center"
    Click="ButtonClick">Click Me!</Button>
...
```

When you specify a value for this attribute, the compiler creates a variable whose value is the UI control that was created from the XAML element. This means you can supplement the XAML configuration of your controls with C# code or change the configuration of elements programmatically. In Listing 2-8, I change the configuration of the button whose name is `FirstButton` in the `ButtonClick` method.

Listing 2-8. *Configuring a Control in Code in Response to an Event*

```
...
private void ButtonClick(object sender, RoutedEventArgs e) {

    FirstButton.Content = "Pressed";
    FirstButton.FontSize = 50;

    System.Diagnostics.Debug.WriteLine("Button Clicked: "
        + ((Button)e.OriginalSource).Content);
}
...
```

I don't have to qualify the control name in any way. In this example, I change the contents of the button and the size of the font. Since these new statements are in the `Click` event handler function, clicking either button will cause the configuration of the `FirstButton` to change. That's all you need to know about XAML for the moment. To summarize:

- XAML is converted into code and merged with the contents of the code-behind file to create a single .NET class.
- You can configure UI controls in XAML or in code.
- Using XAML lets you use the Visual Studio design tools, which are pretty good.

XAML may strike you as verbose and hard to read at first, but you will soon get used to it. I find that it is a lot easier to work with XAML than using just C# code, although I admit it took me quite some time with XAML before I decided that was the case.

Running and Debugging a Metro App

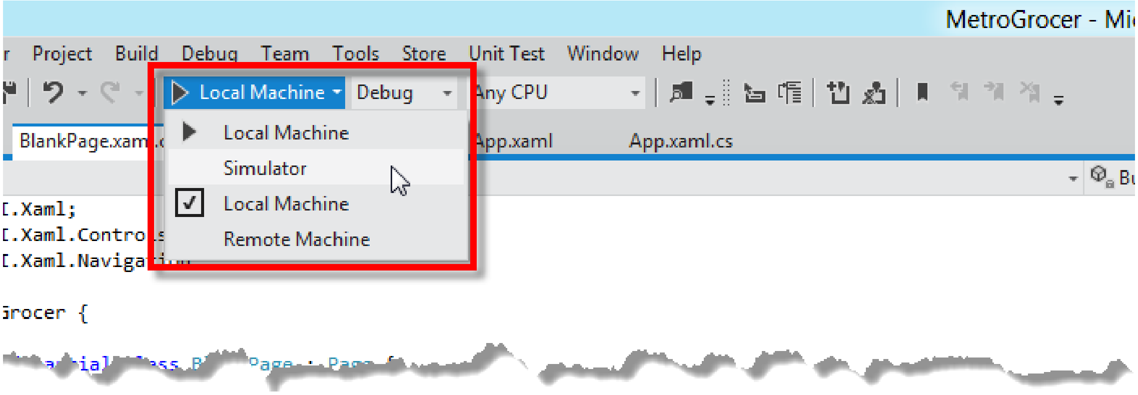
Now that we have a very simple Metro app, it is time to focus on how to run and debug a Metro app. Visual Studio provides three ways to run a Metro app: on the local machine, on the simulator, or on a remote machine.

The problem with the local machine is that development PCs are rarely configured the way that user devices are. Unless you are targeting your app at people with similar spec platforms, then testing on the local machine doesn't give you a representative view of how your application can behave.

Testing on a remote machine is the best approach, but only if you have a range of machines with different capabilities to test with, which is difficult at this point, given that so few Windows 8 machines are available (although I have a small Dell laptop that runs the Consumer Preview of Windows 8 quite happily and lets me debug touchscreen issues).

■ **Tip** You need to download and install the *Remote Tools for Visual Studio 11 Beta* from `http://preview.windows.com` on each remote machine you want to test an app on.

The best compromise is the Visual Studio *simulator*, which provides faithful representation of the Metro experience and lets you change the capabilities of the device you are simulating, including changing the screen size, simulating touch events, and generating synthetic geolocation data. To select the simulator, locate the button on the Visual Studio toolbar that currently says Local Machine, and click the small downward arrow just to the right of it. Select Simulator from the pop-up menu, as shown in Figure 2-5.



```
[.Xaml;
[.Xaml.Contro
[.Xaml.Naviga

Grocer {
```

***Figure 2-5.** Selecting the Visual Studio simulator to test Metro apps*

Running a Metro App in the Simulator

To start the example app, click the toolbar button (which will now say Simulator), or select Start with Debug from the Debug menu. Visual Studio will start the simulator and build and deploy the Metro app, as show in Figure 2-6.

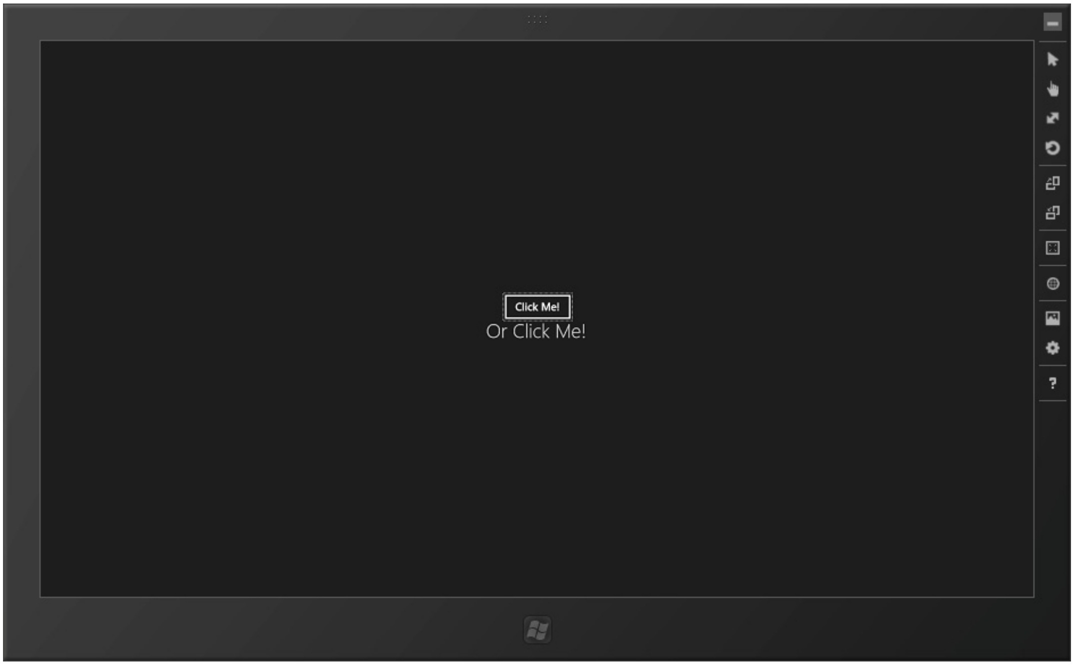


Figure 2-6. *Using the Visual Studio simulator*

You can use the buttons on the right side of the simulator to change the screen size and orientation, switch between mouse and touch input, and synthesize geolocation data. There isn't much to see because the example app is very simple at the moment.

Click either of the buttons on the app layout to trigger the event handler method, and change the configuration of the button. Figure 2-7 shows the result.

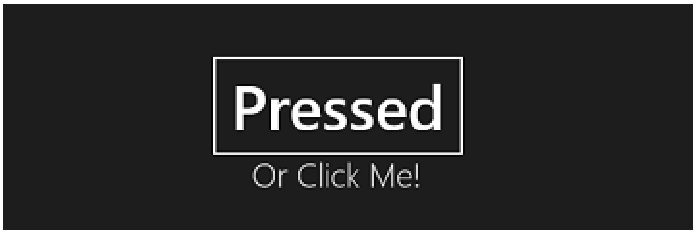


Figure 2-7. *The result of clicking either of the buttons in the example app*

You will also see a message in the Output window, similar to the following:

```
Button Clicked: Pressed
```

Since the Metro app is running in the debugger, any exceptions will cause the debugger to break, allowing you to step through the code just as you would a regular C# project. You can force the debugger to break by setting breakpoints in the source code; aside from the use of the simulator, running and debugging a Metro app uses the standard Visual Studio facilities.

■ **Tip** The simulator works by creating another session on your development machine. This means you can end up with desktop software in two places and can cause the simulator to switch from your Metro app to the desktop. If that happens, you can restart the debugger to reload your Metro app or use the simulator to navigate back to the app itself.

Summary

In this chapter, I introduced the basics of a Metro app written using XAML and C#. I provided a very basic overview of XAML and showed you how it can be applied to create a simple example app.

CHAPTER 3

Testing in Visual Studio

This chapter was submitted by Hrishikesh Dewan

In this chapter, we cover testing in Visual Studio. First, we provide a general testing overview, then walk-through a unit tests example in Visual Studio, and conclude with a walk-through example on Test Driven Development (TDD).

Testing Overview

Before we get started with Visual Studio testing, I will present a short overview of testing. So if you have sufficient background, you may want to skip the first three or four pages. Software testing is the process of validating and verifying that a software program/application/product meets the requirements, works as expected, makes business sense and can be implemented with the same characteristics. Depending on the testing method employed, it can be implemented at any time in the development process. However, most of the test effort traditionally occurred after defining the requirements and completion of coding. However it has clearly been shown that fixing a bug is less expensive when found earlier in the development process. In the Agile approach most of the test effort is, conversely, on-going and early, the testing methodology is defined by the SDLC method adopted. The primary purpose of testing is to detect software failures so that defects may be discovered and corrected. It cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions. Information derived from software testing may be used to correct the process by which software is developed.

Now, let's discuss certain types of testing that are used for specific results.

Functional vs. Non-Functional Testing

Functional testing refers to activities that verify a specific action or function of the code. These are usually found in the code requirements documentation. Functional tests tend to answer the questions "Can the user do this?" or "Does this particular feature work?"

Non-functional testing refers to aspects of the software that may not be related to a specific function or user action, such as scalability or other performance, behavior under certain constraints, or security. Testing will determine the flake point, where extremes of scalability or performance lead to unstable execution.

Static vs. Dynamic Testing

There are many approaches to software testing. Reviews, walkthroughs, or inspections are considered as static testing. Whereas actually executing programmed code with a given set of test cases is referred to as dynamic testing. Static testing can omitted. Dynamic testing generally takes place when the program itself is used for the first time.

The Box Approach

Software testing methods are traditionally divided into white- and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

Black-box Testing

It treats the software as a "black box"—without any knowledge of internal implementation. There are several black-box testing methods:

- Equivalence partitioning
- Boundary value analysis
- All-pairs testing
- Fuzz testing
- Model-based testing
- Exploratory testing
- Specification-based testing

The black-box tester has no "bonds" with the code, and a tester's perception is very simple: code must have bugs. Using the principle, "Ask and you shall receive," black-box testers find bugs where programmers often do not. However it has a bias of being a blind folded exploration and hence is not a very reliable method.

White-box Testing

When the tester has access to the internal data structures and algorithms including the code that implements these, the tester can use multiple techniques:

- *API testing:* Testing of the application using public and private APIs

- *Code coverage:* Creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
- *Fault injection methods:* Improving the coverage of a test by introducing faults to test code paths
- *Mutation testing methods*
- *Static testing:* All types

White-box testing methods can also be used to evaluate the completeness of a test suite that was created with black-box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested

Grey-box Testing

It usually involves having knowledge of internal data structures and algorithms for purposes of designing tests, while executing those tests at the user, or black-box level. The tester is not required to have full access to the software's source code.

By knowing the underlying concepts of how the software works, the tester makes better-informed testing choices while testing the software from outside. Typically, a grey-box tester will be permitted to set up his testing environment; for instance, seeding a database; and the tester can observe the state of the product being tested after performing certain actions.

Visual Testing

The aim of visual testing is to provide developers with the ability to examine what was happening at the point of software failure by presenting the data in such a way that the developer can easily find the information he requires, when the information is expressed clearly.

At the core of visual testing is the idea that showing someone a problem (or a test failure), rather than just describing it, greatly increases clarity and understanding.

Visual testing provides a number of advantages. The quality of communication is increased dramatically because testers can show the problem (and the events leading up to it) to the developer as opposed to just describing it and the need to replicate test failures will cease to exist in many cases. The developer will have all the evidence he requires of a test failure and can instead focus on the cause of the fault and how it should be fixed.

Visual testing is particularly well-suited for environments that deploy agile methods in their development of software, since agile methods require greater communication between testers and developers and collaboration within small teams.

According to the waterfall development model a typical testing cycle goes through the following phases:

- *Requirements analysis:* Testing should begin in the requirements phase of the software development life cycle. During the design phase, testers work with developers in determining what aspects of a design are testable and with what parameters those tests work.

- *Test planning:* Test strategy, test plan, testbed creation. Since many activities will be carried out during testing, a plan is needed.

- *Test development:* Test procedures, test scenarios, test cases, test datasets, test scripts to use in testing software.

- *Test execution:* Testers execute the software based on the plans and test documents then report any errors found to the development team.

- *Test reporting:* Once testing is completed, testers generate metrics and make final reports on their test effort and whether or not the software tested is ready for release.

- *Test result analysis:* Or Defect Analysis, is done by the development team usually along with the client, in order to decide what defects should be assigned, fixed, rejected (i.e. found software working properly) or deferred to be dealt with later.

- *Defect retesting:* Once a defect has been dealt with by the development team, it is retested by the testing team. It is also known as resolution testing.

- *Regression testing:* It is common to have a small test program built of a subset of tests, for each iteration of new, modified, or fixed software, in order to ensure that the latest delivery has not ruined anything, and that the software product as a whole is still working correctly.

- *Test Closure:* Once the test meets the exit criteria, the activities such as capturing the key outputs, lessons learned, results, logs, documents related to the project are archived and used as a reference for future projects.

Why Test?

Testing is the bread and butter for all software developers, and without it, software is nothing but a piece of weird non-linear code which has no guarantee of success in producing the desired output. In this chapter, we will browse over some of the interesting tools and frameworks provided by Visual Studio to test our applications.

There are many applications that have caused miserable output and mishaps in a wide variety of situations. Faults in for instance the *PathFinder Mars software* and other embedded software apps have caused the loss of billions of dollars. For software, testing may not be the guarantee, but the better the test environment and the tests, the better the chance that problems of this variety will not happen.

Testing provides the ability to actually quantify what is being asked for (in user requirements) and what is being delivered. In several cases, testing amounts to quantifying the quality of software too! While, there are several different techniques for writing and executing tests, most software however requires passing verification tests as well. In the parlance of verification, we run software to go through a theoretical model and let mathematical equations and logic analyzes the several different behaviors of the code you have written. Symbolic model checking, a widely accepted technique is being specially used for doing that. While verification is an essential step toward guaranteeing the ultimate success of software, we will not talk about it here. This chapter covers the practical tests only.

In testing, we generally opt to test various parts of software and like the different cycles of software development, testing also applies in several stages of development. Interestingly, Visual Studio provides us a number of tools to aid in each stage of this development life cycle. For example, using Visual Studio you can test your UI, run regression, load tests and even create test plans for your entire software project. We discuss below the several different tests that developers and testers generally opt for and their different types.

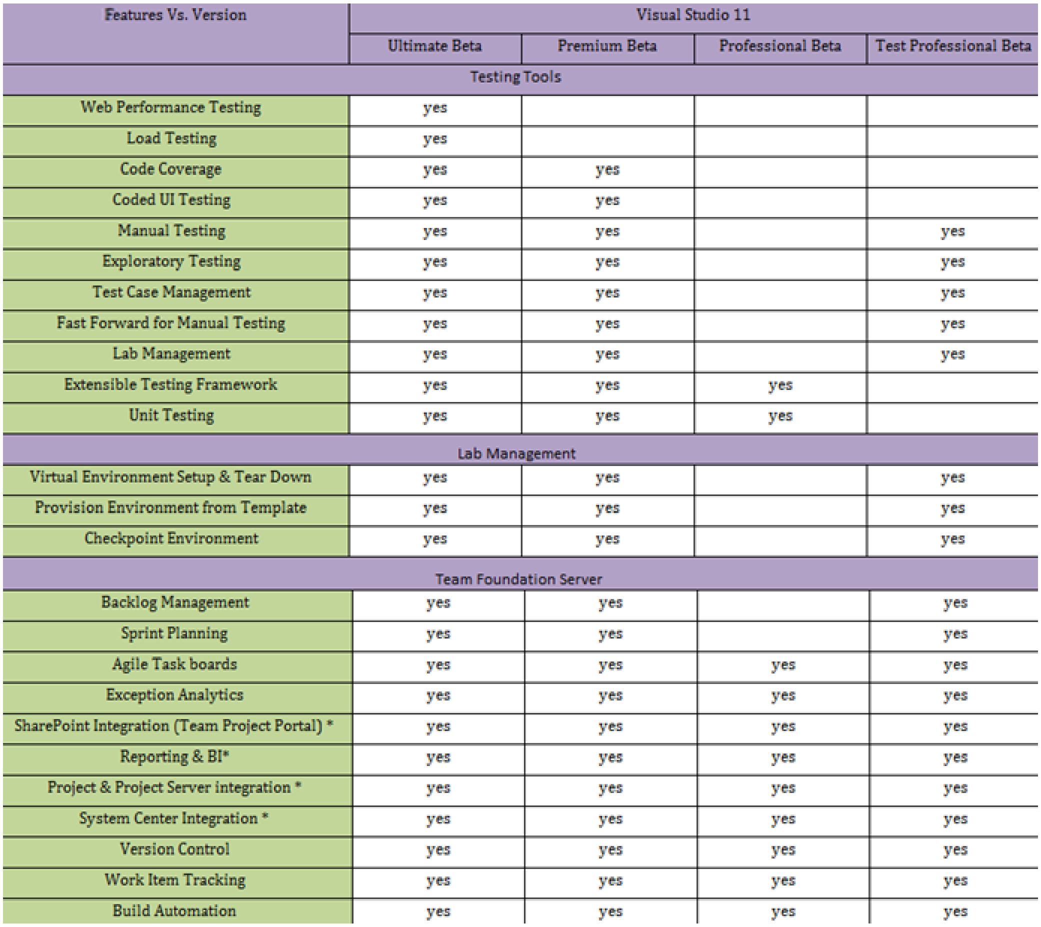
In Visual Studio we have white, black and grey box testing. In white box testing we have the liberty of having access to the source code and we can test individual functionalities associated with it. Unit tests that we developers write, for example, are white box testing. In black box testing, we don't have access to the source code and we tests the software from its functional points where as in grey box testing we do have access to the underlying source code but we do not test from every function or functional point of view, but rather we check most of the time the internal state of the software. Integration testing is a form of grey box testing.

Unit testing is used by developers to check units of code. For a procedural language, units of code could be methods whereas in the case of object oriented development, units of code could be an object or methods included inside a method. Similarly, there are UI tests, which are either done manually or automatically to determine the several different UI issues. Regression tests enable developers to determine code changes by repeatedly running old tests to determine if there is a change in the test output. Its roots are basically to determine the impact of the addition of new code or changes in an existing code base to determine if there is a change in functionality. Load testing enables you to test to check for load conditions in software which might be a part of a functional and/or non-functional requirement of your project and so on.

Visual Studio in its new incarnation provides a lot of functionality to do all of these tests and several more. For example, using Visual Studio, you create an entire test plan, test ability matrix and sequence ordered and unordered tests for your entire project. However, it is important to note that all not all versions of Visual Studio provide all of these functionalities. In some, only a few are supported. In the Professional editions and in Ultimate editions a number of other tests are also possible. Since, this chapter covers testing in general; we stick to the basic level of testing that can be done using Visual Studio and introduce you to a few recent advancements in the theory and practice of testing software. Table 3-1 shows the several different tests that are possible in Visual Studio as of now.

Of the above different test types and suites, Lab Management and Microsoft Test Manager require the Team Foundation Server. Using either of these two test suites, a lot of complex software scenarios can be tested. For example, for testing large distributed applications that involve a number of virtual instances requires Lab tests where Microsoft Test Manager can be effectively used to manage a large number of tests in every build of your software development life cycle.

Table 3-1. *Types and Test and support for the same in the Visual Studio environment. Items marked with a star are sold separately.*



Features Vs. Version	Visual Studio 11			
	Ultimate Beta	Premium Beta	Professional Beta	Test Professional Beta
Testing Tools				
Web Performance Testing	yes			
Load Testing	yes			
Code Coverage	yes	yes		
Coded UI Testing	yes	yes		
Manual Testing	yes	yes		yes
Exploratory Testing	yes	yes		yes
Test Case Management	yes	yes		yes
Fast Forward for Manual Testing	yes	yes		yes
Lab Management	yes	yes		yes
Extensible Testing Framework	yes	yes	yes	
Unit Testing	yes	yes	yes	
Lab Management				
Virtual Environment Setup & Tear Down	yes	yes		yes
Provision Environment from Template	yes	yes		yes
Checkpoint Environment	yes	yes		yes
Team Foundation Server				
Backlog Management	yes	yes		yes
Sprint Planning	yes	yes		yes
Agile Task boards	yes	yes	yes	yes
Exception Analytics	yes	yes	yes	yes
SharePoint Integration (Team Project Portal) *	yes	yes	yes	yes
Reporting & BI*	yes	yes	yes	yes
Project & Project Server integration *	yes	yes	yes	yes
System Center Integration *	yes	yes	yes	yes
Version Control	yes	yes	yes	yes
Work Item Tracking	yes	yes	yes	yes
Build Automation	yes	yes	yes	yes

Unit Test in Visual Studio

As described before, there are a large variety of tools and frameworks from Microsoft as well as other third party vendors for integration with Visual Studio. With the integration of the right set of tools Visual Studio gives the ultimate experience of running and executing tests. The basic professional edition supports writing *unit tests* and hence we cover unit in this section. Note that writing unit tests for your .NET code in Visual Studio can be done using other frameworks as well. Nunit is one such popular framework.

The following example uses a fictitious employee in a particular organization. An employee is a person and hence in the class hierarchy it is derived from a person object. Further, we have methods defined in the employee class that help in increasing, decreasing salaries of the employees, add beneficiaries and so on. Figure 3-1 is the class diagram of the interacting classes. Our purpose is to test to see if at all the methods that are defined in the employee class work or not. In unit test, this is what we precisely do; we as developers write test cases for each individual unit such as methods, objects etc and ascertain whether they really work as we intended or not.

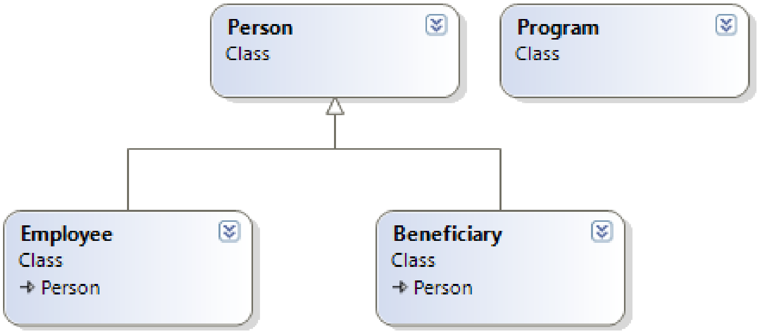


Figure 3-1. *The class diagram fictitious employee class in an organization. The class diagram is generated from Visual Studio and it is yet another interesting tool for developers.*

Listing 3-1. *Source code of the employee class and its associated types*

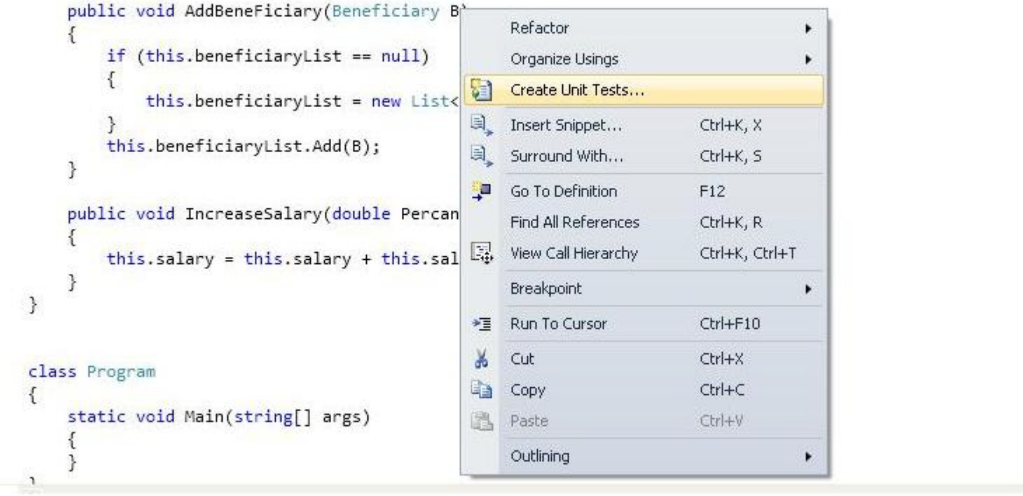
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace test4
{

    public class Person
    {
        string name;

        public string Name
        {
            get { return name; }
            set { name = value; }
        }
        int age;

        public int Age
        {
            get { return age; }
            set { age = value; }
        }

    }

    public class Beneficiary: Person
    {
        string relationship;

        public string Relationship
        {
            get { return relationship; }
            set { relationship = value; }
        }
    }

    public class Employee :Person
    {
        double salary;

        public double Salary
        {
            get { return salary; }
            set { salary = value; }
        }

        List<Beneficiary> beneficiaryList;

        public List<Beneficiary> BeneficiaryList
        {
            get { return beneficiaryList; }
            set { beneficiaryList = value; }
        }



        public void AddBeneFiciary(Beneficiary B)
        {
            if (this.beneficiaryList == null)
            {
                this.beneficiaryList = new List<Beneficiary>();
            }
            this.beneficiaryList.Add(B);
        }

        public void IncreaseSalary(double Percentage)
        {
            this.salary = this.salary + this.salary * Percentage;
        }
    }
}
```

To generate the unit test cases, we just need to follow a few basic steps. I will explain the basic steps one by one and perhaps, you can also try creating your own. Coding can only be learned by practicing and hence it's better if you fire up your Visual Studio now. Although I have included the code for this fictitious organization, it's ideal, if you just imagine and craft a few classes on your own. This will give a firsthand account of why things work in a certain way and how to do it. Remember, testing requires a lot of visualization and thinking and test cases are good enough only if they cover all parts of the code. This is not trivial and requires significant forethought.

Assuming, VS is opened and the above code section as shown in Listing 3-1 is included, generation of test cases and execution requires a few mouse clicks only. To write test cases, the first thing that you have to do is to determine what methods etc you would like to generate test cases for.

So, let's first generate a test case for the *IncreaseSalary* method as defined in the Employee class. This method is trivial; it just adds a percentage to the existing salary of the employee and updates the internal variable as defined in the class. As a developer, we might be interested to ascertain the fact that the salary increase is correctly updated or not. In other words, we are interested in validating our mathematical expression `this.salary = this.salary + this.salary * Percentage`'.

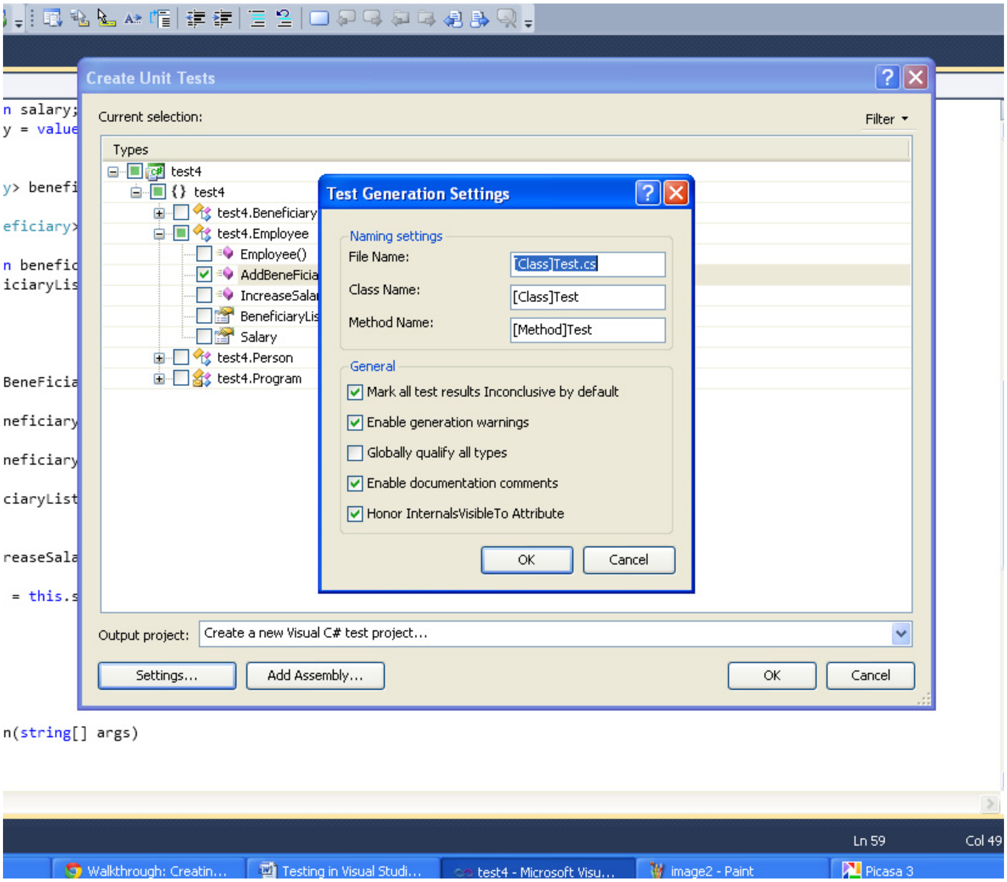
1. Select a method and right click on top of it. Select the Create Unit Tests option from the context menu. It is shown as an example in Figure 3-2. Although, we selected only a single method, the next step in fact would provide us a lot of flexibility.



```
    public void AddBeneFiciary(Beneficiary B
    {
        if (this.beneficiaryList == null)
        {
            this.beneficiaryList = new List<
        }
        this.beneficiaryList.Add(B);
    }

    public void IncreaseSalary(double Percan
    {
        this.salary = this.salary + this.sal
    }
}

class Program
{
    static void Main(string[] args)
    {
    }
```

Figure 3-2. *Selection of a method to create a set of unit tests*

2. Click on Create Test. Figure 3-3 shows the effect of this. This dialog box provides a lot of flexibility in that it allows you to define the name of the test, types, etc. For our purpose, we can just leave it as is. Note that *AddBeneficiary* is automatically added and there are check boxes describing each of the properties, public methods and event private methods as well. We can select as many methods as we want and Visual Studio Automated Code generators automatically add unit test code for each one of them.


Create Unit Tests

Current selection:

Types

- ☑ test4
  - ☑ {} test4
    - □ test4.Beneficiary
    - ☑ test4.Employee
      - □ Employee()
      - ☑ AddBeneFicia
      - □ IncreaseSalar
      - □ BeneficiaryLis
      - □ Salary
    - □ test4.Person
    - □ test4.Program

Test Generation Settings

Naming settings

File Name: [Class]Test.cs

Class Name: [Class]Test

Method Name: [Method]Test

General

☑ Mark all test results Inconclusive by default

☑ Enable generation warnings

□ Globally qualify all types

☑ Enable documentation comments

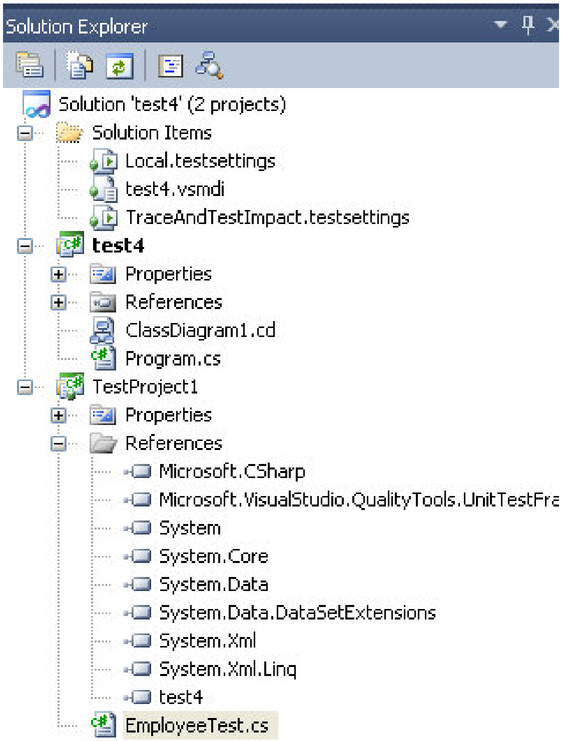
☑ Honor InternalsVisibleTo Attribute

Output project: Create a new Visual C# test project…

Figure 3-3. *The Create Unit Tests and Test Generation Settings dialog boxes*

3. Select the *IncreaseSalary* method and click on *OK* to proceed. A new dialog box comes up. It includes a default name for the project which you can override with your name. We will keep it as is and proceed by creating using the Create button.

The solution window in Figure 3-4 now includes one more project apart from the project that we created. Go over it and check the files; there is a load of C# code that is being generated. Though, most code is generated by the Visual Studio itself, our job is not yet complete. We will still have to do some more modifications before we ask our favorite ID Visual Studio to run or execute these tests for us.



```
Solution 'test4' (2 projects)
  Solution Items
    Local.testsettings
    test4.vsmdi
    TraceAndTestImpact.testsettings
  test4
    Properties
    References
    ClassDiagram1.cd
    Program.cs
  TestProject1
    Properties
    References
      Microsoft.CSharp
      Microsoft.VisualStudio.QualityTools.UnitTestFra
      System
      System.Core
      System.Data
      System.Data.DataSetExtensions
      System.Xml
      System.Xml.Linq
      test4
    EmployeeTest.cs
```

Figure 3-4. *The new solution window.*

So, let's proceed to the solution window and double click on the only .cs file created for us. Double clicking the file opens up a small section of code. If you did everything I prescribed, this is what (Listing 3-2) should look like.

Listing 3-2. *Code generated by the Unit test tool for the classes as shown in Listing 3-1*

```
[TestClass()]
public class EmployeeTest
```

```
{


    private TestContext testContextInstance;


    public TestContext TestContext
    {
        get
        {
            return testContextInstance;
        }
        set
        {
            testContextInstance = value;
        }
    }



    [TestMethod()]
    public void AddBeneFiciaryTest()
    {
        Employee target = new Employee(); // TODO: Initialize to an appropriate value
        target.Name = "yourname";

        Beneficiary B = null; // TODO: Initialize to an appropriate value
        target.AddBeneFiciary(B);
        Assert.Inconclusive("A method that does not return a value cannot be verified.");
    }

    /// <summary>
    ///A test for IncreaseSalary
    ///</summary>
    [TestMethod()]
    public void IncreaseSalaryTest()
    {
        Employee target = new Employee(); // TODO: Initialize to an appropriate value
        double Percentage = 0F; // TODO: Initialize to an appropriate value
        target.IncreaseSalary(Percentage);
        Assert.Inconclusive("A method that does not return a value cannot be verified.");
    }
}
```

Notice in the code segment, there are two essential markers for you from the Visual Studio team at Microsoft: the two TO-DO's. These TO-DO's are the only step that we need to work on. The rest of the things can be left as is unless we are geared toward doing something esoteric.

The first T0-DO asks you to change the way the employee object is constructed. In this case, we need not touch anything. Then we change the value of the percentage to something meaningful, say 10%. So we update the variable to 10. Next we assign 100 to the salary property. And finally, we call the method and use an assert statement to find out if the value required is actually being generated by the code or not. Listing 3-3 shows the code segment of the new test method. Copy as is and you are done. And yes, don't forget to comment out the static *Assert.InConclusive* method. This method leads to test failure and is included just as a demonstration. For developers like us the Assert type is a good friend and there are several different assert methods at our disposal. I would have liked to include a host of them and explain to you their benefits, but this is a Visual Studio chapter and so let me skip them at the interest of the context of the content. But you should make it a point to check on them just after completing this section.

Listing 3-3. *Modified Test Code*

```
 [TestMethod()]
        public void IncreaseSalaryTest()
        {
            Employee target = new Employee(); // TODO: Initialize to an appropriate value

            double Percentage = 10; // TODO: Initialize to an appropriate value
            target.Salary = 100;
            target.IncreaseSalary(Percentage);
            Assert.AreEqual(System.Convert.ToDouble(110), target.Salary);
           // Assert.Inconclusive("A method that does not return a value cannot be
verified.");
        }
```

Okay, we are done, so let’s now proceed to execute this test. You know, it’s simple. Just click on the Test menu in VS (shown as Figure 3-5) and hover over till you get Run or Debug. Choose run and hit “Run All Tests”. The unit test gets executed automatically. Easy isn’t it. It is this easy only with Visual Studio. However, note that the *AddBeneficiary* test fails and your able mind must have by now deduced the facts as to why this test failed. Quite simply, we did not do anything to pass this test.

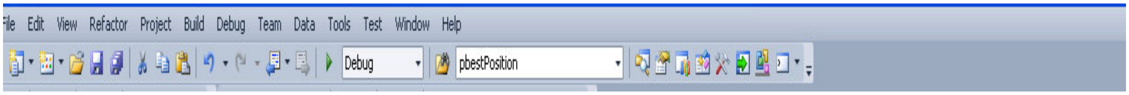
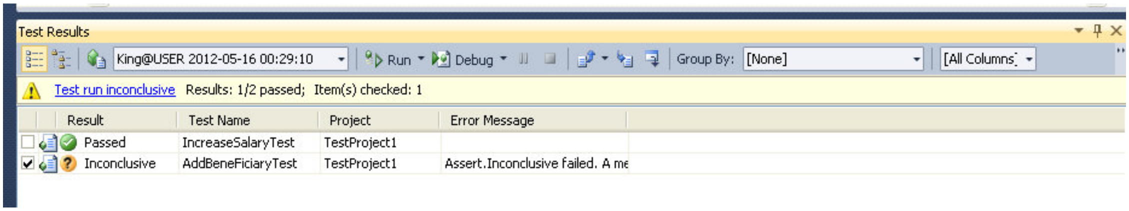


Figure 3-5. *The test menu in the Visual Studio menu.*



Test run inconclusive Results: 1/2 passed; Item(s) checked: 1

	Result	Test Name	Project	Error Message
□	Passed	IncreaseSalaryTest	TestProject1	
☑	Inconclusive	AddBeneFiciaryTest	TestProject1	Assert.Inconclusive failed. A me

Figure 3-6. *Results of running the tests, only the test output is being shown.*

What I have shown you is just a bird’s eye view. Visual Studio Unit test framework can do many more things than shown here.

Test Driven Development

The approach that we just undertook above is a code first-run test-next approach. However, there is an alternative to that as well. In TDD or Test Driven Development, we first write the test cases and then we write code that passes these test cases. In other words, TDD essentially does the above approach, just inside out. The great advantage of TDD is that it allows developers to focus on what is required rather than how it is developed and thereby ensures correctly working software. Visual Studio does not directly support TDD, but there are ways in which we can achieve that. In the next paragraphs, we are exactly going to see how this can be done.

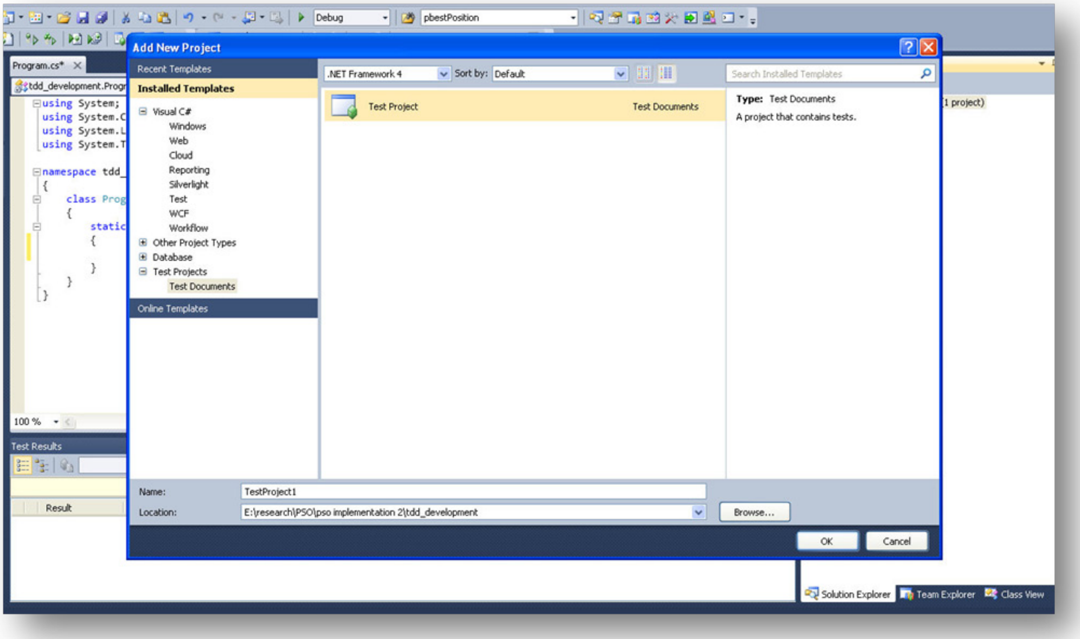
The premise of a TDD approach is to first specify what you want to develop. For example, in our last example as noted in Listing 3-1, we would like to have an employee type his basic properties such as Name, Age and Beneficiaries (for maybe some tax schemes etc). Further, we would like to have a method wherein we could increase the salary of the employee. Clearly when an employee object is created, all of the properties checked are null or void. The system is expected to add or update this after an object is created. In TDD, we specifically list these requirements and we start by writing test code instead of the actual employee class. The technique that is being followed is: Red-Green-Refactor; that is we first write a test, allow it to fail and then create the necessary code section that allows it to pass and then finally when everything is set, we clean our re-factor code to make it elegant.

As usual, the best way to learn on how code works is start by writing code; we follow a very simple technique:

1. The first step in creating a TDD-based development is to create a project itself. In this case, let us create a console application “tdd1”.

2. Once the basic console application is created, we now desire to write a few unit tests. That is, we want to follow the Red of the Red-Green-Refactor rule.

In Step 2, unlike Step 1, we create a test project. Test projects can be found under the Test project section of Visual Studio templates. Select it and click Create. A new test project will be created. This is the same place we would write unit test code. Figure 3-7 shows the selection procedure in case you are un-able to find it.



.NET Framework 4 Sort by: Default

Type: Test Documents

A project that contains tests.

Name: TestProject1

Location: E:\research\PSO\pso implementation 2\tdd_development

Figure 3-7. Selecting a new test project for test driven development.

Note that after execution of Step 2, a new set of files will be created. Of particular importance is the UnitText.cs class. This class has a single class defined as shown in Listing 3-4.

Listing 3-4. The VS generated UnitTest type and the corresponding method.

```
[TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestMethod1()
        {
        }
    }
```

In this class, we have method TestMethod1(). The name of the class and the method is not intuitive and hence we may choose to rename it as per our needs. For example, in this case, we may rename the type to EmployeeTest and EmployeeInitializationTest method. In other words, using this unit test we would like to know if the employee object that would be created has the salary field blank when it is first created. Listing 3-5 shows the changed code.

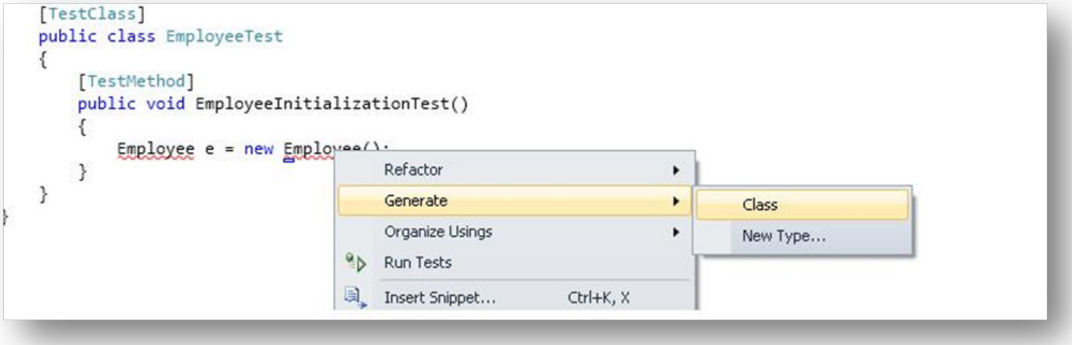
Listing 1.5. *Renaming of the Test class and the method.*

```
[TestClass]
   public class EmployeeTest
   {
       [TestMethod]
       public void EmployeeInitializationTest()
       {

       }
   }
```

At this stage, we don’t have anything to test. Recall from our past example, where in the first steps we created an object of target type and then we set the properties and so on. In this case, we will do the same thing; however the difference is that we don’t have anything to fill.

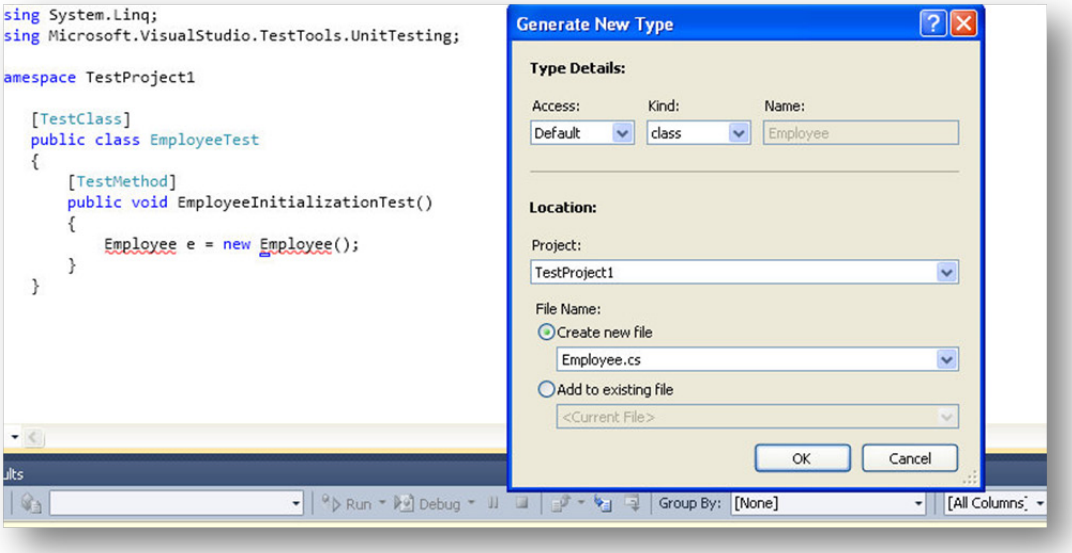
At this step, we have two different paths and therefore we will mark Step 3a and Step 3b.

In Step 3b inside the EmployeeInitializationTest method we simply try to create an Employee object method. We know it won’t compile, but we ask Visual Studio to come in to help us. First, complete the line and right click over the type. The screen is shown in Figure 3-8. Click on Generate or click on New Type. The Generate method creates a class in the Test Project itself, whereas in the case of new type, you have the option of choosing access modifiers, which project, and so on. We will show the screen shots of both of them in Figure 3-8 and Figure 3-9. For our use case, we will choose ‘New Type’ instead and create a public type in our original console application.



```
 [TestClass]
 public class EmployeeTest
 {
     [TestMethod]
     public void EmployeeInitializationTest()
     {
         Employee e = new Employee();
     }
 }
}
```

Figure 3-8. *Generating the corresponding type using VS context menu*



```
sing System.Linq;
sing Microsoft.VisualStudio.TestTools.UnitTesting;

amespace TestProject1

    [TestClass]
    public class EmployeeTest
    {
        [TestMethod]
        public void EmployeeInitializationTest()
        {
            Employee e = new Employee();
        }
    }
```

Generate New Type

Type Details:

Access: Default

Kind: class

Name: Employee

Location:

Project: TestProject1

File Name:

● Create new file

Employee.cs

○ Add to existing file

<Current File>

Figure 3-9. Generating the corresponding type using ‘new type’ using the VS Context menu. The dialog box shows the screen which allows you to generate a type in a desired project etc.

If you have followed the steps as noted, you will find that the employee class is generated in the console application. Notice the beauty in that we are not required to write a single line of code.

Okay, at this stage, we can move a little ahead and then write the actual code for asserting whether a new generated employee object has the salary component ‘0’ as desired. We are insisting on ‘0’ because we wish to have salary as a double and hence by default it will be initialized to 0. Listing 3-6 shows the new test method.

Listing 3-6. Modified method that includes the assert statement

```
[TestMethod]
public void EmployeeInitializationTest()
{
    Employee e = new Employee();
    Assert.IsNull(e.Salary);

}
```

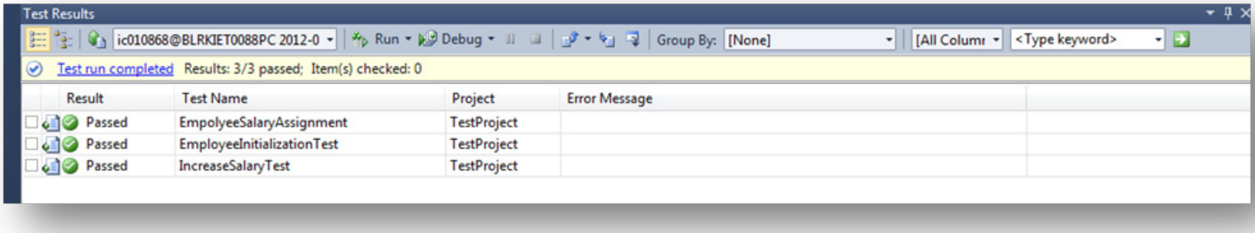
However, note that although we intend or wish to work on the employee salary field, there is no such entity created so far. VS is also of great help in this scenario. Like before, just right click and select generate property and you will find that a new property has been created. There is a small hitch however. If you browse the source code, you will find that the property so created is of type object. In this case, we can simply remove the keyword object and replace it with a double. We can continue doing this for the rest of the properties and methods and I list them below the source code for you to see and implement. Do this and you will have your first iteration complete. Run the test now by using the same context sensitive menu. The previous Red now shows up to Green. At this point, you can move to the third step and re-factor the code. But there's nothing that needs to be re-factored at this point. You safely skip to the rest of the sections by creating a few more tests and move on to create stubs to fulfill them.

Listing 3-7. *Unit Test Method*

```
 [TestClass]
    public class EmployeeTest
    {
        [TestMethod]
        public void EmployeeNameValidationTest()
        {
            Employee e = new Employee();
            Assert.IsNotNull(e.Name);
        }


        [TestMethod]
        public void EmployeeInitializationTest()
        {
            Employee e = new Employee();
            Assert.AreEqual(System.Convert.ToDouble(0), e.Salary);

        }

        [TestMethod]
        public void EmpolyeeSalaryAssignment()
        {
            Employee e = new Employee();
            e.Salary = 100;
            Assert.AreEqual(System.Convert.ToDouble(100), e.Salary);
        }

        [TestMethod]
        public void IncreaseSalaryTest()
        {
            Employee e = new Employee();
            e.Salary = 100;
            Assert.AreEqual(System.Convert.ToDouble(110),e.IncreaseSalary(10));
        }
}
```

Listing 3-8. *The Employee class so generated*

```
public class Employee
    {
        public double Salary { get; set; }
        public double IncreaseSalary(double p)
        {
            return this.Salary = this.Salary + (this.Salary * p) / 100;

        }
        public string Name { get; set; }
    }
```



ic010868@BLRKIET0088PC 2012-0

Group By: [None]

[All Colum

<Type keyword>

Test run completed Results: 3/3 passed; Item(s) checked: 0

Result	Test Name	Project	Error Message
□ Passed	EmpolyeeSalaryAssignment	TestProject	
□ Passed	EmployeeInitializationTest	TestProject	
□ Passed	IncreaseSalaryTest	TestProject	

Figure 3-10. *Passed Test results*

Summary

Although we have talked a bit about different testing techniques and shown ways of doing that in C# using Visual Studio, there are a lot of other things that can be done using this wonderful IDE. For example, using your programming skills you can adapt the usual way of testing unit tests, introduce several new test harnesses and even modify several intrinsic properties of the test framework. However, be warned of one thing -- testing does have its limitations. For example, if you are writing a random or rather a pseudo random number generator, these kinds of tests like regression, load etc and will never let you ascertain if you have really written a pseudo random generator. Statistical tests are required to check whether a sequence of bits is random or not. NIST (National Institute for Standard) for example provides a statistical test suite to determine if a sequence of bits so generated is really random or not. I hope I have left you with enough material to move forward with your VS IDE, and check your code to see if it is really working as per your expectations!

CHAPTER 4

Introduction to Application Lifecycle Management

Adapted from Chapter 2 of Pro Application Lifecycle Management with Visual Studio, 2/e

What do you think about when you hear the term Application Lifecycle Management, or ALM as it is usually referred to? During a seminar tour in 2005 in Sweden presenting Visual Studio Team System, we asked people what ALM was and whether they cared about it. To our surprise, many people equated ALM with operations and maintenance. This is still often the case today when we visit companies.

Maybe that was your answer as well? Doesn't ALM include more than just operations? Yes, it does. ALM is the thread that ties together the development lifecycle. It involves all the steps necessary to coordinate the development lifecycle activities. Operations are just one part of the ALM process. This chapter provides an overview of that process and sets the stage for discussions on making effective ALM assessments in Chapter 5 and using Team Foundation Server (TFS) to support the ALM process in Chapter 6.

Roles in the ALM Process

All software development includes various steps performed by people playing specific roles in the process. There are many different roles or perhaps we could call them disciplines in the ALM process, and I define some of them in this section. (Please note that the process could include more roles, but I have tried to focus on the main ones.) Take a look at Figure 4-1, which illustrates ALM and some of its roles.

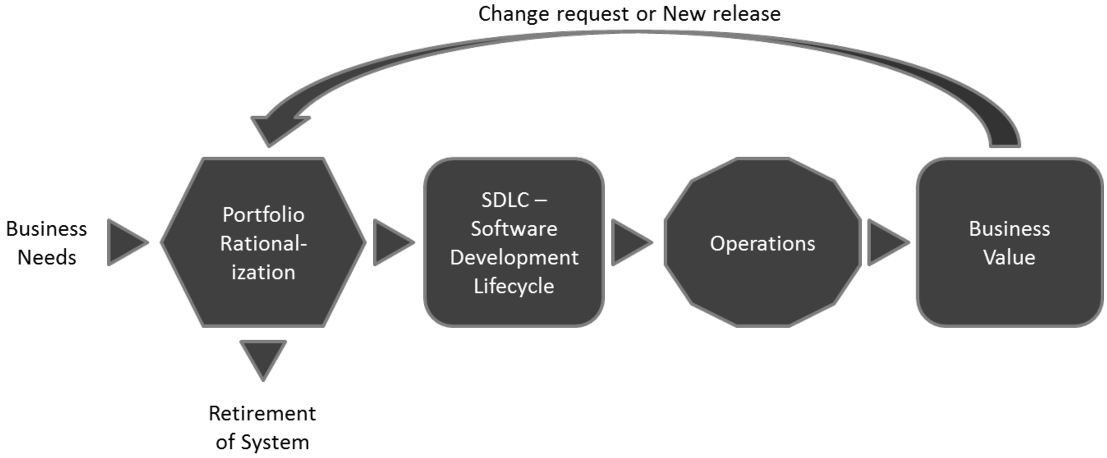

Change request or New release
Business Needs
Portfolio Rational-ization
SDLC – Software Development Lifecycle
Operations
Business Value
Retirement of System

Figure 4-1. *The Application Lifecycle Management process and some of the roles in it*

It is essential to understand that all business software development is a team effort. The roles collaborate on projects in order to deliver business value to the organization. If we don't have this collaboration, the value of the system most likely will be considerably lower than it could be. If we look at it one step up from the actual project level, it is also important to have collaboration between all roles involved in the ALM process, so that we perform this process in the most optimal way possible. The roles in the ALM process include the following:

- *Business manager*: Somebody has to make the decision that a development activity is going to start. After initial analysis of the business needs, a business manager decides to initiate a project for the development of an application or system that will deliver the expected business value. A business manager, for instance, will have to be involved in the approval process for the new suggested project, including portfolio rationalization, before a decision to go ahead is reached. Other people involved in this process are of course IT managers, because the IT staff will probably be involved in the project's development and deployment into the infrastructure.

- *Project manager, Product owner or scrum master*: Suitable individuals are selected to fill these roles and set to work on the project after the decision to go ahead is made. Ideally, these people continue leading the project all the way through, so that we have continuity in project management.

- *Project Management Office (PMO) decision makers*: These individuals are also involved in planning because a new project might very well change or expand the company's portfolio.

- *Business analyst*: After requirements collection starts, the business analyst has much to do. A business analyst is responsible for analyzing the business needs and requirements of the stakeholders, to help identify business problems and propose solutions. Within the systems development lifecycle, the business analyst typically performs a collaborative function between the business side of an enterprise and the providers of services to the enterprise.
- *Architect*: The architect starts drawing the initial picture of the solution. I will not go into great detail here because Chapter 5 does that. But briefly, the architect draws the blueprint of the system, and the system designers or engineers use this blueprint. This blueprint includes the level of freedom necessary in the system. For instance scalability, hardware replacement, new user interfaces, and so on. All of these issues must be considered by the architect.
- *User Experience (UX) design team*: Hopefully UX design is a core deliverable and not something we leave to the developers to handle. UX design is sadly overlooked and should definitely have more consideration. It is important to have close collaboration between the UX team (which could be just one person) and the development team. The best solution is obviously to have a UX expert in the development team all through the project, but that is sometimes not possible. The UX design is so important in making sure users can really perceive the value of the system. We can write the best business logic in the world, but if the UX is badly designed, the users will probably never think the system is any good.
- *Database administrators (DBAs)*: Almost every business system or application uses a database in some way. The DBAs are the ones who can make our databases run like lightning with good up-time, so it is essential to use their expertise in any project involving a database. Be nice to them; they can give you lots of tips on how to make a smarter system.
- *Developers*: Developers, developers, developers as Microsoft CEO Steve Ballmer shouted in a famous video. And who can blame him? These are the people doing their magic to realize the system that we are building by using the architecture blueprint drawn from the requirements. Moreover, these are the people who have to modify or extend the code when change requests come in.
- *Test*: I would rather not see testing as a separate activity. Testing is something we should consider from the first time we write down a requirement, and continue doing during the whole process.
- *Operations and maintenance staff*: Here it is. When an application or system is finished, it is handed over to operations. The operations staff takes care of it until it retires, often with the help of the original developers who come in a do bug fixes and new upgrades. Don't forget to involve these people early in the process, at the point when the initial architecture is considered, and keep them in the project until all is done. They can give great input as to what can and can't be done in the company infrastructure. So operations is just one part, but an important one, of ALM.

All project efforts are done as collaborative work. No role can act separate from any of the others if we are to succeed with any project. It is essential for everybody in a project to have a collaborative mindset and to have the business value as the primary focus at every phase of the project.

If you are part of an agile project like a scrum project, you might have only three roles: product owner, scrum master and team members. This does not mean that roles described above do not apply, though! They are all essential in most projects; it's just that in an agile project you may not be labeled a developer or an architect. Rather, you are there as a team member and as such you and your co-members share responsibility for the work you have committed to. Although the agile world is outside the scope of this chapter, the complete book, *Pro Application Lifecycle Management with Visual Studio*, covers it in some depth.

Four Ways of Looking at Application Lifecycle Management

Application Lifecycle Management is the glue that ties together all these roles and the activities they perform. Let's consider four ways of looking at ALM (see Figure 4-2). We have chosen these four because we have seen this separation in so many of the organizations I have worked with or spoken to:

- *Software Development Lifecycle (SDLC) view*: This is perhaps the most common way of looking at ALM because development has "owned" management of the application lifecycle for a long time. This could very well be an effect of the gap between the business side and the IT side in most organizations, and IT has taken the lead on this.

- *Service management or operations view*: Operations have also had (in our experience) an unfortunate separation from IT development. This has resulted in operations having their own view of ALM and the problems in this area.

- *Application Portfolio Management (APM) view*: Again maybe because of the gap between business and IT, I have seen many organizations with a portfolio ALM strategy in which IT development is only one small part. From a business view, the focus has been on how to handle the portfolio itself and not on the whole ALM process.

- *Chief information officer (CIO) view (or the unified view)*: Fortunately, some organizations focus on the whole ALM process by including all three of the preceding views. This is the only way to take control over, and optimize, ALM. For a CIO, it is essential to have this view all the time; otherwise, things can easily get out of hand.

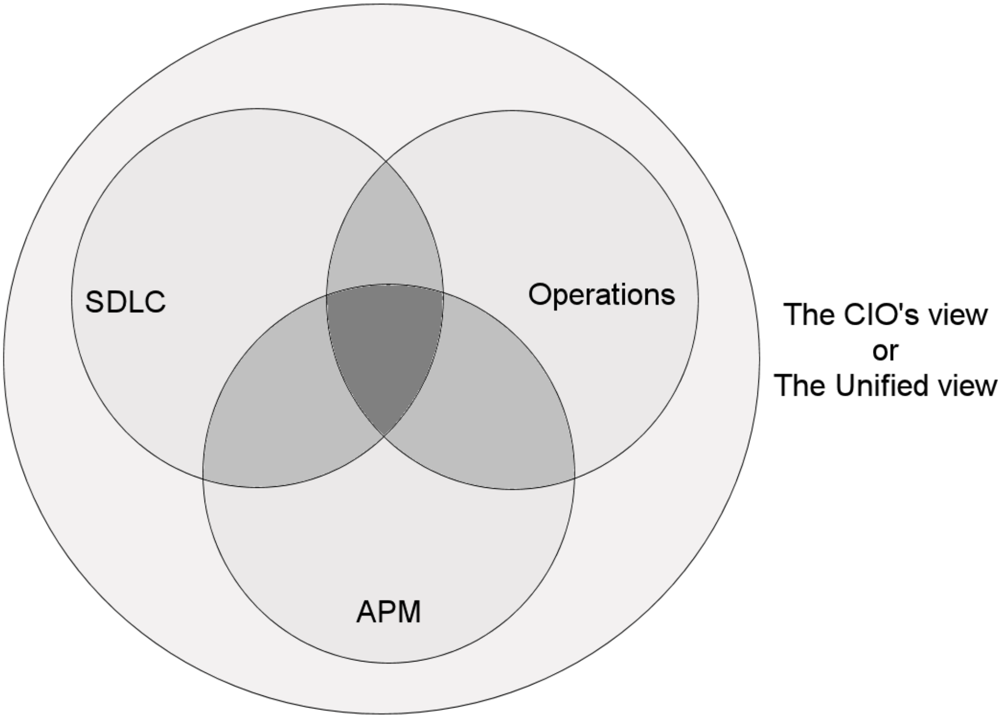


Figure 4-2. *The four ways of looking at ALM*

The SDLC View

Let's take a look at ALM from an SDLC perspective first. In Figure 4-3, you can see the different phases of a typical development project. Keep in mind that this is just a simplified view for the sake of this discussion. We have also tried to fit in the different roles from the ALM process presented earlier.

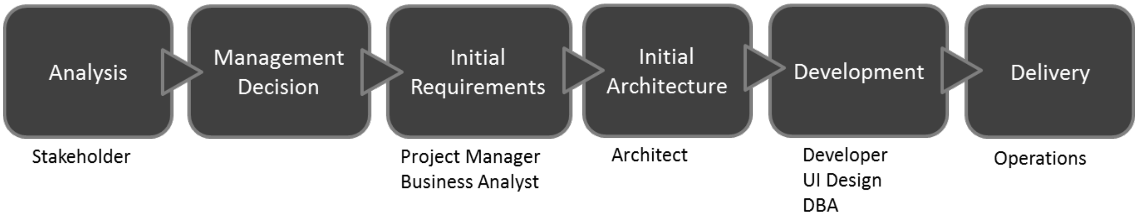


Figure 4-3. *A simplified view of a typical development project*

First, somebody comes up with an idea based on an analysis of the business needs: “Hey, wouldn’t it be great if we had a system that could help us do this (whatever the idea is)?” It could also be the other way around: the idea comes first, and the business value is evaluated based on the idea.

So, an analysis or feasibility study is performed, costs are estimated, and hopefully a decision is made by IT and business management to start the project as an IT project. A project manager (PM) is selected to be responsible for the project and starts gathering requirements with the help of business analysts, PMO decision makers, and users or others affected. The PM also starts planning the project in as much detail as possible at this moment.

When that is done, the architect starts looking at how to realize the new system, and the initial design is chosen. The initial design is then evaluated and changed based on what happens in the project and what happens with requirements all through the project. After that, the development starts, including work performed by developers, user interface (UI) designers, and DBAs (and any other person not mentioned here but who is important for the project).

Testing is, at least for us, something done all along the way—from requirements specification to delivered code—so this not a separate box in Figure 4-3. After the system has gone through acceptance testing, it is delivered to operations for use in the organization. Of course it doesn’t end here. This cycle is most often repeated over and over again as new versions are rolled out and bug fixes implemented.

What ALM does in this development process is support the coordination of all development lifecycle activities from the preceding process through the following:[1]

- Enforcement of processes that span these activities.
- Management of relationships between development artifacts used or produced by these activities. (In other words, we talk about traceability here.)
- Reporting on progress of the development effort as a whole.

As you can see from this, ALM does not support a specific activity in itself. Its purpose is to keep all activities in sync. It does this just so we can focus on delivering systems that meet the needs and requirements of the business. By having an ALM process helping us synchronize our development activities, we can more easily see if any activity is underperforming and thus more easily take corrective actions.

The Service Management or Operations View

From a service management or operations view, we can look at ALM as in this quote from *ITIL Application Management* by the Office of Government Commerce in United Kingdom (TSO, 2002): ALM “focuses on the activities that are involved with the deployment, operation, support, and optimization of the application. The main objective is to ensure that the application, once built and deployed, can meet the service level that has been defined for it.”

When we see ALM from this perspective, it focuses on the life of an application or system in a production environment. If in the SDLC view the development lifecycle started with the decision to go ahead with the project, here it starts with deployment into the production environment. Once deployed, the application is operated by the operations crew. Bug fixes and change requests are handled by them and they also pat it on its back to make it feel good and run smoothly.

[1] “The Changing Faces of Application Lifecycle Management,” Forrester, 2006.

This is a quite healthy way of looking at ALM in our opinion, because we think that both development and operations are two pieces of ALM, cooperating in order to manage the whole ALM process. Both pieces are also something that should be thought of when planning a development project from the beginning; we cannot have one without the other.

The Application Portfolio Management View

The third view we will look at is the Application Portfolio Management (APM) view of ALM. In this view, we see the application as a product managed as part of a portfolio of products. We can say that APM is a subset of Project Portfolio Management (PPM). Figure 4-4 describes this process.

This view comes from the Project Management Institute (PMI). Managing resources and the projects they work on is very important for any organization. In this view, we can see that the product lifecycle starts with a business plan—the product is the application or system that is one part of the business plan. An idea for an application is turned into a project and carried out through the project phases, until it is turned over to operations as a finished product.

When business requirements change or a new release (an upgrade in this figure) is required for some other reason, the project lifecycle starts again, and a new release is handed over to operations. After a while (maybe years) the system or application is discarded (this is called *divestment*, the opposite of investment). This view does not specifically speak about the operations part or the development part but should be seen in the light of APM instead.

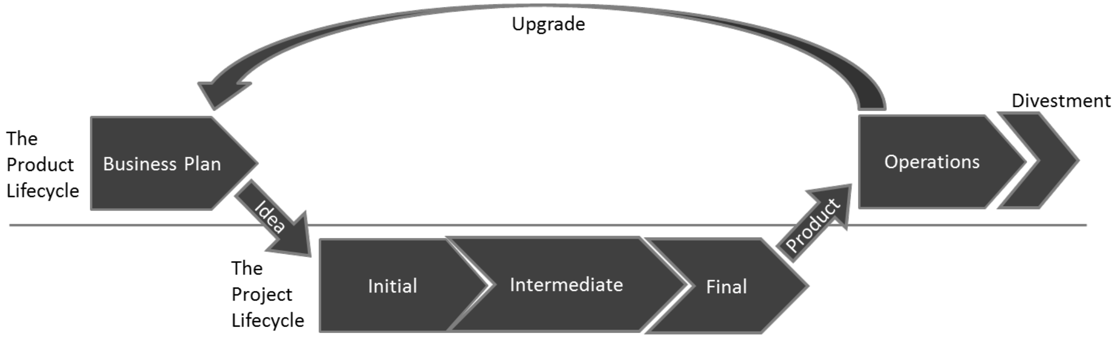


Figure 4-4. *The PMI view of ALM*

The Unified View

The final view is a unified view of ALM. In this view, we have made an effort to align all these views with the business. Here we do as the CIO would do: we focus on the business needs, not on separate views. This we do to improve the capacity and agility of the project from start to end. Figure 4-5 shows an overview of how these three views are included in the whole unified ALM aspect of a business.

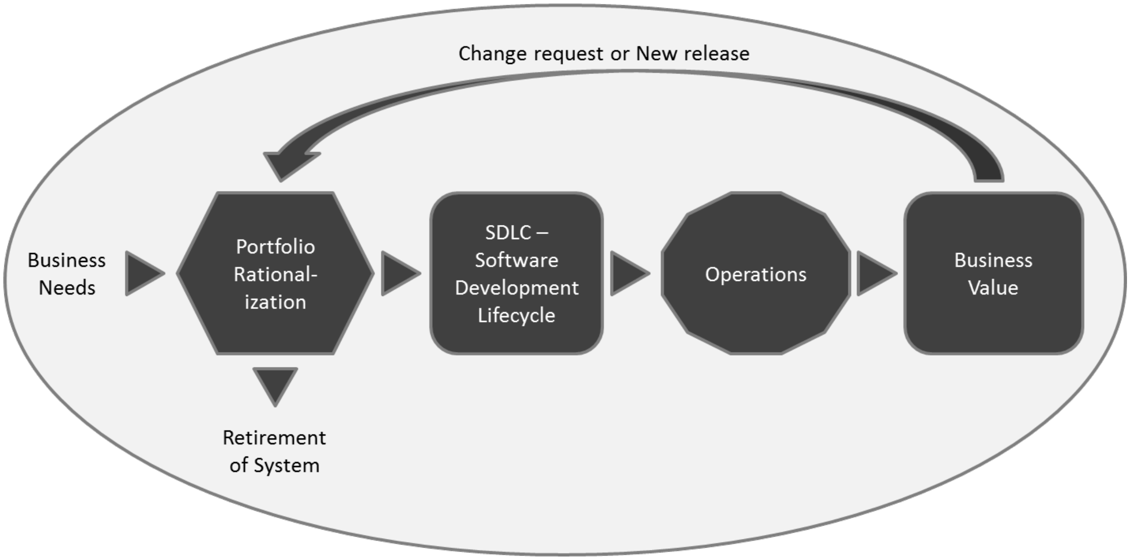

Change request or New release
Business Needs
Portfolio Rational-ization
SDLC – Software Development Lifecycle
Operations
Business Value
Retirement of System

Figure 4-5. *The CIO's view takes into consideration all three views previously mentioned.*

Three Pillars of Application Lifecycle Management

Let's now look at some important pillars we find in ALM, independent of the view we take. Forrester Research defines three pillars in ALM,[2] as shown in Figure 4-6.

[2] "The Changing Faces of Application Lifecycle Management," Forrester, 2006.

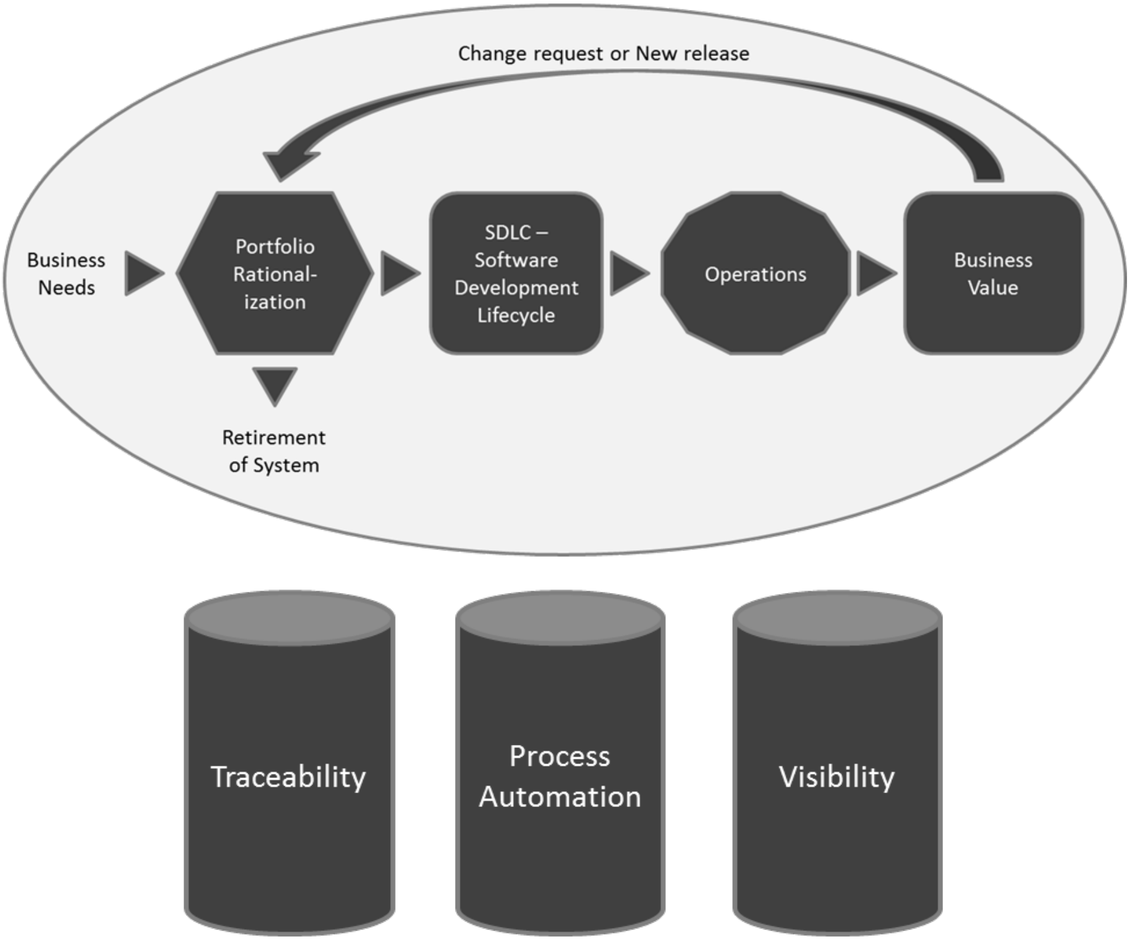


Figure 4-6. *The three pillars of ALM*

Traceability of Relationships Between Artifacts

Imagine for a moment a customer who has stopped doing upgrades on systems that are running in production because the company has poor or even no traceability in its systems. To this client it is far too expensive to do upgrades because of the unexpected effects even a small change could have. The company has no way of knowing which original requirements were implemented where in the applications. This customer claims, and we have seen and heard this in discussions with many other customers, that traceability can be a major cost driver in any enterprise if not done correctly.

There must be a way of tracing the requirements all the way to delivered code—through architect models, design models, build scripts, unit tests, test cases, and so on—not only to make it easier to go back into the system when implementing bug fixes, but also to demonstrate that the system has delivered the things the business wanted.

Another reason for traceability is internal as well as external compliance with rules and regulations. If we develop applications for the medical industry, for example, we need to have compliance with FDA regulations. We also need to have traceability when change requests are coming in so that we know where we updated the system and in which version we performed the update.

Automation of High-Level Processes

The next pillar is automation of high-level processes. All organizations have processes. For example, there are approval processes to control hand-offs between the analysis and design or build steps, or between deployment and testing. Much of this is done manually in many projects, and ALM stresses the importance of automating these tasks for a more effective and less time-consuming process. Having an automated process also decreases the error rate compared to handling the process manually.

Visibility into the Progress of Development Efforts

The third and last pillar is providing visibility into the progress of development efforts. Many managers and stakeholders have limited visibility into the progress of our development projects. The visibility they have often comes from steering group meetings, during which the project manager reviews the current situation. Some would argue that this limitation is good, but if we want to have an effective process, we must ensure visibility.

Other interest groups such as project members also have limited visibility of the whole project despite being part of the project. This often comes from the fact that reporting is hard to do and often involves a lot of manual work. Daily status reports would quite simply take too much time and effort to produce, especially when we have information in many repositories.

A Brief History of ALM Tools

We can resolve these three pillars manually if we want, without the use of tools or automation. ALM is not a new process description even though Microsoft, IBM, and the other large players are giving it a lot more visibility. We can, for instance, continue to use Excel spreadsheets, or as one of our most dedicated agile colleagues does, use Post-it notes and a pad of paper, to track requirements through use cases/scenarios, test cases, code, build, and so on to delivered code. It works, but this process takes a lot of time and requires much manual effort. With constant pressure to keep costs down, we need to make tracking requirements more effective.

Of course, project members can simplify the process by keeping reporting to the bare minimum. With a good tool, or set of tools, we can cut time (and thus costs) and effort, and still get the required traceability we want in our projects. The same goes for reporting and all those other activities we have. Tools can, in my opinion, help us be more effective, and also help us automate much of the ALM process right into the tool(s).

By having the process built directly into our tools, it is much easier for the people involved to not miss any important step by simplifying anything. For instance, the agile friend we mentioned could definitely gain much from this, and he has now started looking into Team Foundation Server (TFS) to see how that set of tools can help him and his teams be more productive. So process automation and the use of tools to support and simplify our daily jobs are great things because they can keep us from making unnecessary mistakes.

Take a look at Figure 4-7. It shows the disciplines involved in the ALM process according to Serena Software Inc.

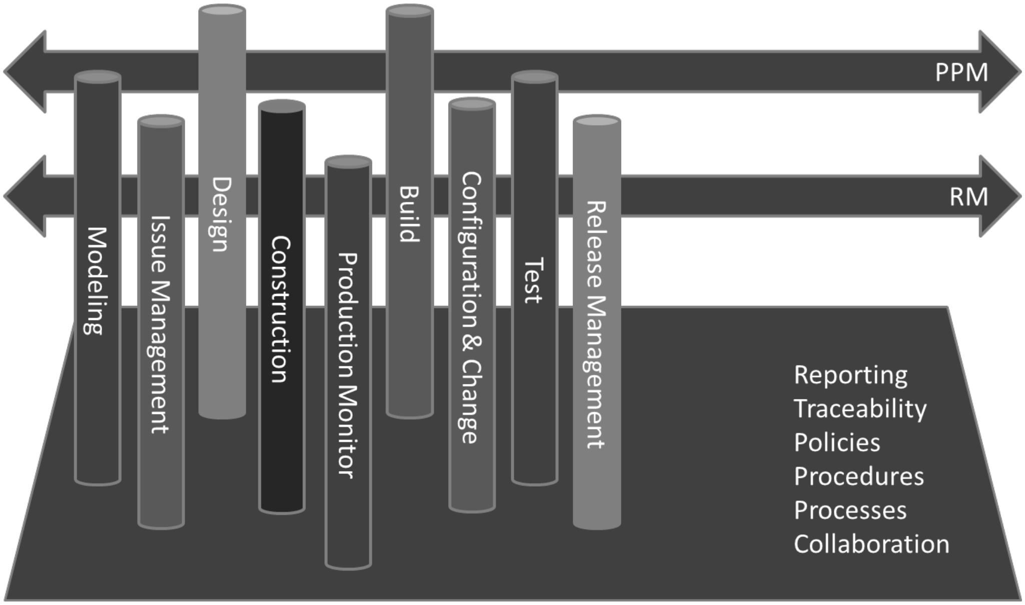


Figure 4-7. *Disciplines in ALM*[3]

There are nine disciplines according to Serena Software:

- *Modeling*: Software modeling
- *Issue management*: Keeping track of the incoming issues during both development and operations
- *Design*: Designing the system or application
- *Construction:* Developing of the system or application
- *Production monitoring*: The work of the operations staff
- *Build*: Building the executable code
- *Configuration and change management*: Keeping track of changes and configuration of our applications
- *Test*: Testing the software
- *Release management*: Planning the releases of our application

[3] Kelly A. Shaw, Ph.D, "Application Lifecycle Management for the Enterprise," Serena Software Inc, April 2007. http://www.serena.com/docs/repository/company/serena_alm_2.0_for_t.pdf)

Imagine the Herculean task of keeping all those things in order manually. That would be impossible if we wanted to get something right and keep an eye on the status of projects. Projects seem to be going better because the number of failed projects is decreasing. Much of this progress is, according to Michael Azoff at the Butler Group,[4] the result of some "major changes in software development: open source software projects; the Agile development movement; and advances in tooling, notably Application Lifecycle Management (ALM) tools." Some of these results are also confirmed by later research, for instance by Scott W. Ambler at Ambysoft (http://www.ambysoft.com/surveys/success2011.html) and Alan Radding (http://resources.devx.com/ibm/Article/47065). Now we understand finding tools and development processes to help us in the ALM process is important.

This has led to increasing awareness of the ALM process among enterprises. We can see this among the customers we have. ALM is much more important now than it was only five years ago.

Application Lifecycle Management 1.0

As software has become more and more complex, role specialization has increased in IT organizations[5]. This has led to functional silos in different areas (roles) such as project management, business analysis, architecture, development, database administration, testing, and so on. As you may recall from the beginning of this chapter, you can see this in the ALM circle. There is no problem with having these silos in a company, but having them without any positive interaction between them is.

There is always a problem when we build great and impenetrable walls around us. Most ALM vendors have driven the wall construction because most of their tools historically have been developed for particular silos. If we look at build management tools, they have supported the build silo (naturally) but have little or no interaction with test and validation tools (which is kind of strange since the first thing that usually happens in a test cycle is the build)—just to mention one area. This occurs despite the fact that interaction between roles can generate obvious synergies that have great potential. We need to synchronize the ALM process to make the role-centric processes a part of the overall process. This might sound obvious, but has just not happened until lately.

Instead of having complete integration between the roles or disciplines mentioned in the beginning of the chapter and the tools they use, we have had point-to-point integration—for example, we could have a development tool slightly integrated with the testing tool (or probably the other way around). Each tool uses its own data repository, so traceability and reporting is hard to handle in such an environment as well (see Figure 4-8).

[4] Michael Azoff, "Application Lifecycle Management Making a Difference," February 2007, Enterprise Networks and Services, OpinionWire.

[5] "The Challenges of Software Change Management in Today's Siloed IT Organizations," a commissioned study conducted by Forrester Consulting on behalf of Serena Software, November 2006.

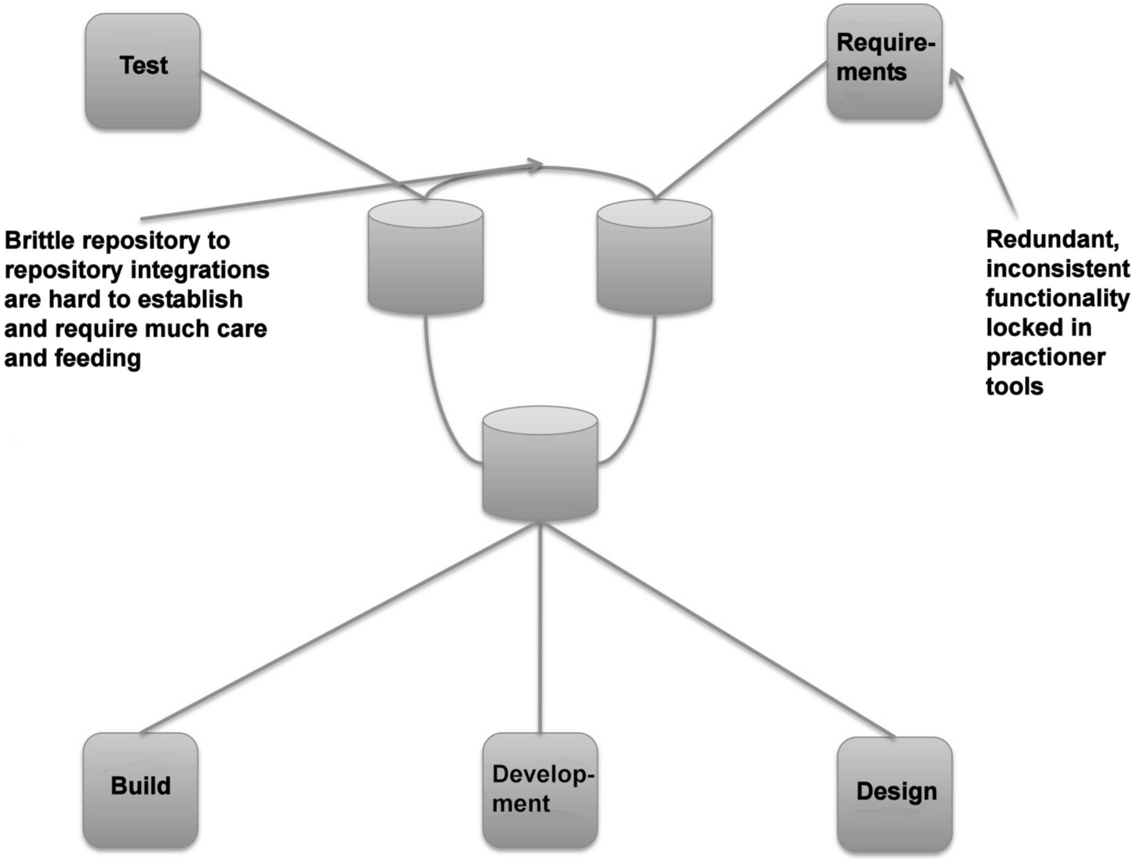


Figure 4-8. ALM 1.0

This point-to-point integration makes the ALM process fragile and quite expensive as well. Just imagine what happens when one tool is updated or replaced. Suddenly, the integration might break and new solutions have to be found to get it working again. This scenario can be reality if, for example, old functions in the updated or replaced tool are obsolete and the new one does not support backward compatibility. This can be hard to solve even with integration between just two tools. Imagine what happens if we have a more complex situation, including several more tools. We have seen projects using six or seven tools during development, creating a fragile solution when new versions have been released.

The tools have also been centered on one discipline. In real life, a project member working as a developer, for instance, quite often also acts as an architect or tester. Because the people in each of these disciplines have their own tool (or set of tools), the project member must use several tools and switch between them. It could also be that the task system is separated from the rest of the tools so to start working on a task, a developer must first retrieve the task from the task system—probably print it out, or copy and paste it, then open the requirements system to check the requirement, then look at the architecture in that system, and finally open the development tool to start working. Hopefully, the testing tools are integrated into the development tool; otherwise, yet another tool must be used. All this switching costs valuable time better put into solving the task.

Having multiple tools for each project member is obviously costly as well because they all need licenses for the tools they use. Even with open source tools that may be free of charge, we have maintenance costs, adaptations of the tools, developer costs, and so on. Maintenance can be very expensive, so we should not forget this even when the tools are free. So, such a scenario can be very costly and very complex. It will also most likely be fragile.

As an example, we have two co-workers working at a large medical company in Gothenburg. They have this mix of tools in their everyday work. We asked them to estimate how much time they needed to switch between tools and get information from one tool to another. They said they probably spend half an hour to an hour each day syncing their work. Most times they are on the lower end of this scale, but still in the long run this is a lot of time and money. My friends also experienced big problems whenever they needed to upgrade any of the systems they used.

One other problem with traditional ALM tools worth mentioning is that vendors often have added features, for example, adapting a test tool to support issue and defect management. And in the issue management system, some features might have been added to support testing. Because neither of the tools have enough features to support both disciplines, the users are confused and will not know which tool to use. In the end, most purchase both just to be safe, and end up with the integration issues described earlier.

Application Lifecycle Management 2.0

So let's take a look at what the emerging tools and practices (including processes and methodologies) in ALM 2.0 try to do for us. According to Forrester Research, the emerging ALM "is a platform for the coordination and management of development activities, not a collection of lifecycle tools with locked-in and limited ALM features."[6] Figure 4-9 and Table 4-1 summarize these efforts.

[6] Carey Schwaber, "The Changing Faces of Application Lifecycle Management," http://i.bnet.com/logos/whitepapers/Serena_Life_Cycle_Management.pdf)

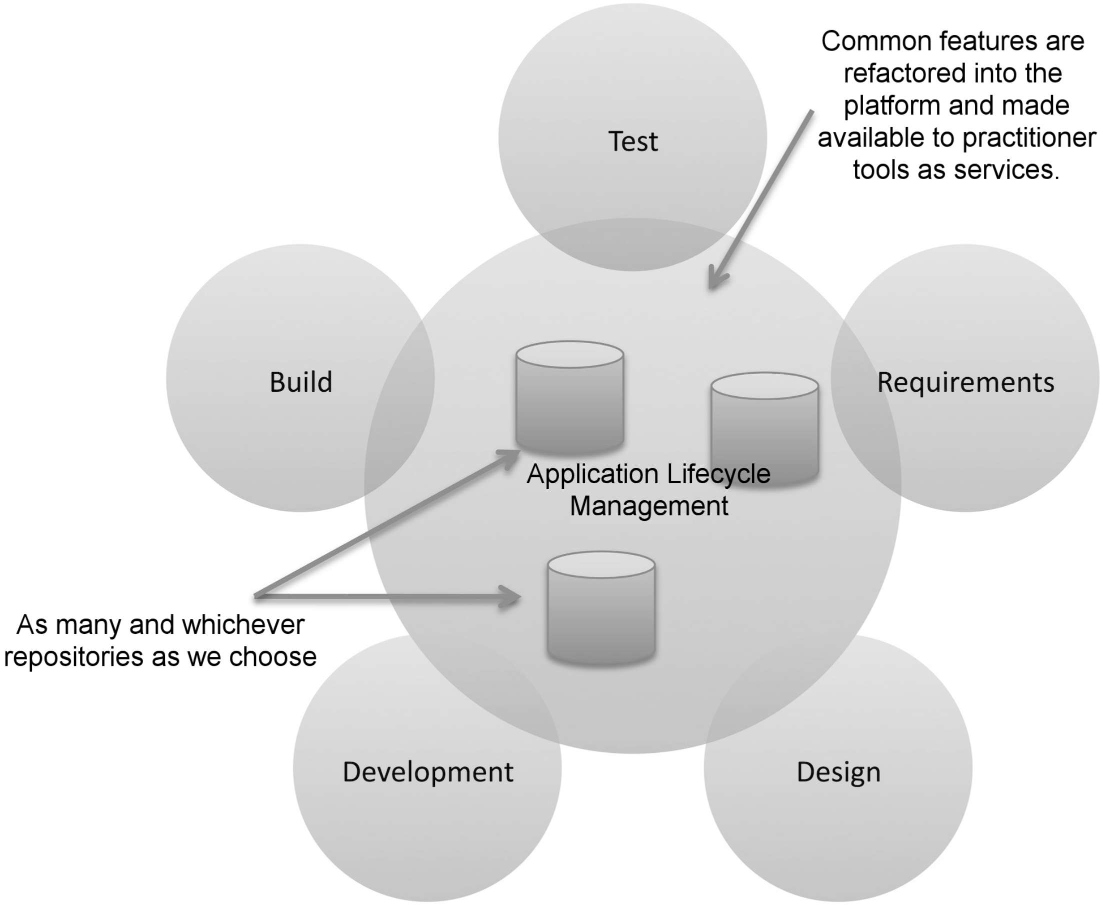

Test
Common features are refactored into the platform and made available to practitioner tools as services.
Build
Requirements
Application Lifecycle Management
As many and whichever repositories as we choose
Development
Design

Figure 4-9. ALM 2.0

Table 4-1. Characteristics in ALM 2.0

Characteristics	Benefit
Practitioner tools assembled out of plug-ins	Customers pay only for the features they need. Practitioners find the features they need faster.
Common services available across practitioner tools	Easier for vendor to deploy enhancements to shared features. Ensure correspondence of activities across practitioner tools.

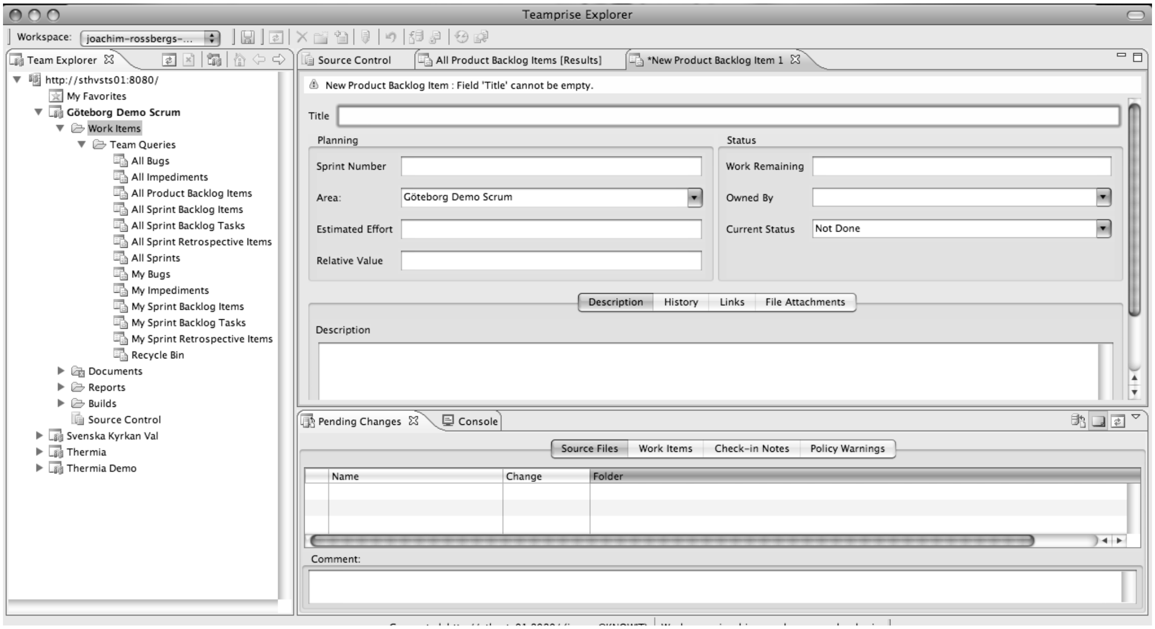
Characteristics	Benefit
Repository neutral	No need to migrate old assets. Better support for cross-platform development.
Use of open integration standards	Easier for customers and partners to build deeper integrations with third-party tools.
Microprocesses and macroprocesses governed by externalized workflow	Processes are versionable assets. Processes can share common components.

One of the first things we can see is a focus on plug-ins. This means that from one tool, we can add the features we need to perform the tasks we want. Without using several tools! If you have used Visual Studio, you have seen that it is quite straightforward to add new plug-ins into this development environment. Support for Windows Communication Foundation (WCF) and Windows Presentation Services, for example, was available as plug-ins long before their support was added as a part of Visual Studio 2008.

Having the plug-in option and making it easy for third-party vendors to write plug-ins for the tool greatly eases the integration problems discussed earlier. You can almost compare this to a smorgasbord, where you choose the things you want. So far this has mostly been adopted by development tool vendors such as Microsoft, but more are coming. Not much has happened outside of development tools so far but TFS offers some nice features that definitely will help us a lot.

■ **Note** Smorgasbord is originally a Swedish term for a big buffet of food where the guests can pick and choose the dishes they want. In Sweden we have smorgasbord at Easter, Christmas, and other holidays.

Teamprise, a third-party vendor, developed a solution giving access to the Team Foundation Server (TFS) from a wide array of platforms, including Mac OS X (see Figure 4-10). In November 2009, Teamprise was acquired by Microsoft. After the acquisition, TFS 2010 changed its name to Team Explorer Everywhere. In writing these ALM chapters, we used Team Explorer Everywhere on our Mac OSX laptops using the Eclipse development platform.



Workspace: joachim-rossbergs-...

New Product Backlog Item : Field 'Title' cannot be empty.

Area: Göteborg Demo Scrum

Current Status: Not Done

Figure 4-10. Team Explorer Everywhere in Mac OS X

Another thing that eases development efforts is that vendors in ALM 2.0 are focusing more on identifying features common to multiple tools and integrating these into the ALM platform. We find things like the following among these:

- Collaboration
- Workflow
- Security
- Reporting and analysis

You will see in Chapter 5 that TFS has these features embedded out of the box. Microsoft uses Microsoft Office SharePoint Server/Windows SharePoint Services as part of the TFS suite for collaboration among team members, for example. Active Directory is used for authentication and security. SQL Server obviously is the data repository giving us access to tools such as SQL Server Analysis Services for analytics and SQL Server Report Builder for reports. Microsoft has done a good job in giving us tools to get the job done.

Forrester Research also mentions that ALM 2.0 should be repository neutral. They say that there should not be a single repository but many, so we are not required to use the storage solution that the vendor proposes. IBM, for example, has declared that their coming ALM solution will integrate with a wide variety of repositories, such as Concurrent Versions System (CVS) and Subversion, just to mention two. This approach removes the obstacle of gathering and synchronizing data, giving us easier access to progress reports, and so on. Microsoft uses an extensive set of web services and plug-ins to solve the same thing. They have one storage center (SQL Server), but by exposing functionality through the use of web services, they have made it fairly easy to connect to other tools as well.

An open and extensible ALM solution lets companies integrate their own choice of repository into the ALM tool. Both Microsoft and IBM have solutions—data warehouse adapters—that enable existing repositories to be tied into the ALM system. It is probable that a large organization that has already made investments in tools and repositories in the past doesn't want to change everything for a new ALM system; hence it is essential to have this option.

Any way we choose to solve the problem will work, giving us possibilities of having a well-connected and synchronized ALM platform.

Furthermore, ALM 2.0 focuses on being built on an open integration standard. As you know, Microsoft exposes TFS functionality through web services. This is not publicly documented however, so we need to do some research and trial and error before we can get this working. This way, we can support new tools as long as they also use an open standard; and third-party vendors have the option of writing tons of cool and productive tools for us.

Process support built into the ALM platform is another important feature. By this we mean having the automated support for the ALM process built right into the tool(s). We can, for instance, have the development process (RUP, SCRUM, XP, and so on) automated in the tool, reminding us of each step in the process so that we don't miss creating and maintaining any deliverables or checkpoints.

In the case of TFS, you will see that this support includes having the document structure, including templates for all documents, available on the project web site, directly after having created a new TFS project. We can also imagine a tool with built-in capabilities helping us in requirements gathering and specification, for instance—letting us add requirements and specs into the tool and have them transformed into tasks assigned to the correct role without having to do this manually.

An organization is not likely to scrap a way of working just because the new ALM tool says it cannot import that specific process. A lot of money has often been invested in developing a process, and an organization is not likely interested in spending the same amount again learning a new one. With ALM 2.0, it is possible to store the ALM process in a readable format such as XML.

The benefits include that the process can be easily modified, version controlled (you need to do this yourself, TFS doesn't do it for you), and reported upon. The ALM platform can then import the process and execute the application development process descriptions in it. Microsoft uses XML to store the development process in TFS. In the process XML file the whole ALM process is described, and many different process files can coexist. This means we can choose which process template we want to base our project on when creating a new project.

As we saw earlier, it is important for an enterprise to have control over its project portfolio to better allocate and control resources. So far, none of the ALM vendors have integrated this support into the ALM platform. There may be good reasons for this though. For instance, while portfolio management may require data from ALM, the reverse is probably not the case.

The good thing is that having a standards-based platform makes integration with PPM tools a lot easier.

Application Lifecycle Management 2.0+

So far not all the ALM 2.0 features have been implemented by tools vendors. There are various reasons for this. One of these is the fact that it is not quite easy for any company to move to a single integrated suite, no matter how promising the benefits might look when you see them. To make such a switch would mean changing the way we work in our development processes and even in our company. Companies have invested in tools and practices, and spending time and money on a new platform can require a lot more investment.

For Microsoft focused development organizations, the switch might not be so hard however, at least not for the developers. They already use Visual Studio, SharePoint, and many other applications in their daily life and the switch is not that big. But Microsoft is not the only platform out there and competitors like IBM, Serena, and HP still have some work to do to convince the market.

We also can see that repository neutral standards and services have not evolved over time. Microsoft's for instance still rely on SQL Server as a repository and have not built in much support for other databases or services. The same goes for most competition to TFS.

■ **Note** Virtually all vendors will use ALM tools to lock in customers to as many of their products as possible—especially *expensive* major strategic products like RDBMS. After all they live mostly on license sales.

The growth of agile development and project management in recent years has also changed the way ALM must support development teams and organizations. We can see a clear change from requirements specs to backlog-driven work, and the tooling we use needs to support this in a good way.

Agile practices such as build and test automation become critical for our ALM tools to support. Test Driven Development (TDD) continues to rise and more and more developers require their tools to support this way of working. If the tools don't do that they will be of little use for an agile organization. Microsoft has really taken the agile way of working to their hearts in the development of TFS. In the complete book, we show you all you need to know about the support for agile practices in TFS.

We can also see a move from traditional project management toward an agile view where the product owner and scrum master require support from the tools as well. Backlog grooming (the art of grooming our requirements in the agile world), agile estimation and planning, reporting—important to these roles—need to be integrated to the overall ALM solution.

The connection between operations and maintenance becomes more and more important. Our ALM tools should integrate with the tools used by these parts of the organization. Fortunately, Microsoft has plans for this. They plan to make the integration between MS Systems Operations Manager and TFS much simpler and built in out of the box, adding support for maintenance and operations.

At the end of 2011 we attended a Microsoft Western Europe ALM gathering in Madrid. At this session Jeffrey Hammond from Forrester drew an image like Figure 4-11. In his talk Hammond extended traditional ALM with what he called ALM 2.0+. Traditional ALM covers traceability, reporting, and process automation. Hammond envisions the future of ALM to also include collaboration and work planning. We like this idea and we will try to show you that Microsoft also has support for this in TFS.

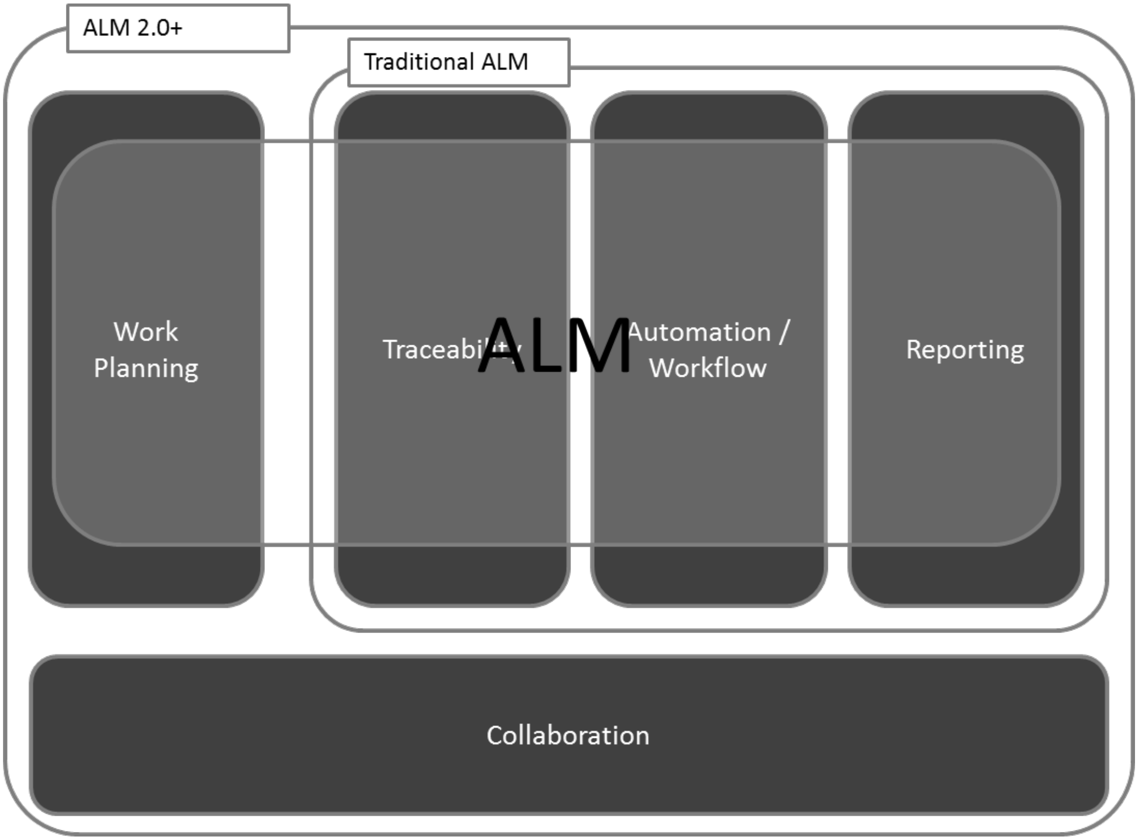


Figure 4-11. *Future ALM according to Jeffrey Hammond at Forrester Research*

ALM and PPM

ALM and PPM can support each other quite well. Data from the ALM repository can be an excellent source of data for the PPM tool, and hence decisions can be based on the results of the PPM tool. This requires a (one-way) working connection between the two, of course. Manual efforts by cutting and pasting information are not good enough because they are slow and error prone. A good integration between the two repositories gives project portfolio decision-makers access to accurate and timely data. This eases their decision-making process.

Gartner identifies five key activities in the PPM decision process that benefit from a working integration:[7]

- Review current projects and their status.
- Review the application portfolio impact on resources (which resources are available and when are they available, for instance).
- Present new capital requests. (Do we need to add more resources to a project?)
- Reprioritize the portfolio.
- Examine investments for effectiveness (basically reviewing the outcome of projects within six months of delivery).

A lot of data, important for these activities, can be found in the ALM repository.

> *"Microsoft's solution to the ALM and PPM integration still has some room for improvement but works very well. "By installing Microsoft Visual Studio Team Foundation Server 2010 and Project Server Integration Feature Pack, project managers can use Project Server to access up-to-date project status and resource availability across agile and formal software teams who work in Team Foundation. This integration enables data to flow from work items in Team Foundation Server to tasks in enterprise project plans in Project Server. With this feature pack, project managers and software development teams can use the tools that they prefer, work at the level of precision that supports their needs, and share information transparently. After the two server products are configured, the synchronization engine maintains scheduling data and resource usage for the configured data in the mapped enterprise project plan and team project."*
>
> http://msdn.microsoft.com/en-us/library/gg455680.aspx

Microsoft also offers the Microsoft Office Enterprise Project Management (EPM) solution (http://www.microsoft.com/project/en-us/solutions.aspx). This is Microsoft's end-to-end collaborative project and portfolio environment. This solution aims to help an organization gain visibility, insight, and to control across all work, enhancing decision making, improving alignment with business strategy, maximizing resource utilization, and measuring and helping to increase operations efficiency. Whoa, lots of words, right? This is taken straight from the Microsoft web site. We will not delve in to the specifics of this solution here but will tell you a little about its three parts.

First, there is the Microsoft Office Project Professional. If you are a project manager, chances are that you know this product already. If not, many of us have definitely seen the Gantt schema the project manager has produced. We can use Project Professional as a stand-alone product if we need it for only single projects. But the real value comes when we connect it to the second part of the Microsoft EPM solution: the Microsoft Office Project Server. This server offers the possibilities of resource management,

[7] "Application Lifecycle Management and PPM," Serena, 2006.

scheduling, reporting, and collaboration capabilities in the Microsoft EPM solution. We use the Project Server to store project and resource information in a central repository.

The third part of Microsoft's EPM suite used to be the Microsoft Office Project Portfolio Server. Since version 2010 this is built in to Microsoft Project Server. This gives us a management solution, enabling organizations to get control of their product portfolios so that they best align with the company business strategies.

All in all this helps us to handle our portfolios, projects, and resources so that we can plan our needs.

Summary

In this chapter, we have described an overview of what ALM aims for and what it takes for the ALM platform to support a healthy ALM process. We have seen that there are four ways of looking at ALM:

- *Software Development Lifecycle (SDLC) view*
- *Service management or operations view*
- *Application Portfolio Management (APM) view*
- *Chief information officer (CIO) view (or the unified view)*

We know that traceability, automation of high-level processes, and visibility into development processes are three pillars of ALM. Other important key components are collaboration, workflow, security, reporting, analytics, being open standards based, being plug-in friendly, and much more, and they are focus areas of ALM. You have also seen that ALM is the coordination and synchronization of all development lifecycle activities.

A good ALM tool should help us implement and automate these pillars and components to deliver better business value to our company or organization.

Chapter 5 explores the value of performing an ALM assessment and considers approaches to that process.

CHAPTER 5

ALM Assessments

Adapted from Chapter 4 of Pro Application Lifecycle Management with Visual Studio, 2/e

Often in our work we need to perform an assessment of a given situation. It could be describing a system's present architecture and then coming up with a plan for improving it, or it could be assessing how a system scales. When we start working with people instead of technology, the focus of the assessments is on a different level: suddenly there is a need to consider human factors and not only technological topics. Before TFS entered the scene, we conducted surveys of how a department functioned, for instance. We did such assessments by interviewing people from the organizations and thereby getting a picture of the situation.

Interviews are complex and result in a lot of information that you need to process afterward. Mostly we prepared by writing questions on a form used for the interviews. Any follow-up questions that came up were carefully documented and included in later assessments. So this form started to become our tool, even though we did not have it in an application. Instead we used a Word document that was updated and printed out for each interview.

However, we have been a bit reluctant to use only tools. We thought about digitizing the questions into a web application or the like and letting the subjects answer the questions themselves, but we didn't want to abandon the interview part. Tools can help, but they can also hinder because you can become too dependent on them. Another aspect of using tools for this purpose is that if we let a single person answer our questions in, let's say a web form, we wouldn't be around to ask the follow-up questions. These questions enable us to learn so much more than we would if we only looked at the answers to the original bunch of questions.

We started working with TFS a few years back when the product was new. We felt that it was a good foundation for taking control of an ALM process. It lacked (and still does in some cases) some of the support we wanted, but it was a good start, and nothing else was offered on the Microsoft platform to compete with it. The more we dived into it, the more we started thinking of how we best could evaluate an organization to implement TFS at their site. We realized very quickly that ALM was an important part of an organization's ability to improve their software development cycle, and TFS was an excellent tool to help our customers. We could see that we had tools for visibility, traceability and automation of high-level processes that fit nicely into the ALM concept.

Microsoft released their ALM assessments on the web some years ago (www.microsoft.com/almassessment), including the Application Platform Optimization (APO) model. Microsoft provided its APO model to help IT organizations understand and adopt a more flexible and agile application platform. These felt like a good start, so we set off to figure out how to best use these tools to help our customers implement TFS. Previously, we found that many of our customers used only a fragment of the true potential of TFS (most often the version-control system). If we could show them why they should use more features, they would get so much more out of it and at the same time be more successful in running their projects.

Microsoft's assessments were tools pretty much like web questionnaires, which is why we set off to work on creating an assessment based on Microsoft's Application Platform Capability Assessments, but added an interview part, as well. The reason for this is that a tool cannot read between the lines. It cannot hear what a person says apart from what is actually answered. It is important to have the capability to ask follow-up questions, to clarify and discuss where needed.

One great advantage of the Microsoft assessments is that after everything is entered into the tool, the tool can handle much of the data processing automatically. This decreases the labor involved in the manual processing of the large amount of data an interview gives you. We "only" need to make sure that we enter the most realistic values into the system, and that's where the interview part can help us out. By using the questions from the online assessment as a form for our interviews, we have a good foundation for interviews. Then we can use the results of the interviews, including follow-up questions and observations, to enter data into the assessments form, giving us the capability to reflect on each question and choose the best answer for the organization we are working with.

Microsoft Application Platform Optimization (APO) Model

APO is a part of Microsoft's Dynamic IT initiative that also includes the Infrastructure Optimization model and the Business Productivity Infrastructure model. There are four primary goals of the Dynamic IT initiative:

- Manage complexity and achieve agility
- Protect information and control access
- Advance the business via IT solutions
- Amplify the impact of your people

These models are aimed at helping customers better understand the current IT capabilities in their organizations and, based on these results, take the capabilities to a higher level of maturity.

In this section, we'll give you a brief overview of the Infrastructure Optimization model and the Business Productivity Infrastructure model before focusing on APO.

Infrastructure Optimization Model

With the Infrastructure Optimization model, Microsoft focuses on four areas: the desktop infrastructure, the server infrastructure, the remote infrastructure (which covers how to handle remote offices or locations), and virtualization. Based on best practices internal to Microsoft, as well as on feedback from customers, Microsoft has provided an approach they say will do three things:

- Control costs by looking over hardware, utilities, and space expenses in the data center to see where we can reduce costs. The costs also can be controlled or reduced by optimizing deployment testing and training expenses, as well as reducing security breaches (in addition to other strategies outside the scope of these chapters).
- Improve service levels. This can be done, for example, by reducing service interruptions caused by security breaches, having a robust disaster recovery strategy, and avoiding desktop configuration conflicts.

- Drive agility. Here we find topics familiar to us by now, such as increasing the ability to adapt to business changes.

Business Productivity Infrastructure Model

Microsoft provides a definition for what optimizing our Business Productivity Infrastructure means:

> *The Business Productivity Infrastructure Optimization (IO) model includes a complete set of technologies that helps streamline the management and control of content, data, and processes across all areas of your business. It helps simplify how people work together, makes processes and content management more efficient, and improves the quality of business insight while enabling IT to increase responsiveness and have a strategic impact on the business.*
>
> *The Business Productivity IO Model defines five capabilities that are required to build a more agile infrastructure:*
>
> - *Collaboration*
> - *Unified Communications*
> - *Enterprise Content Management*
> - *Reporting & Analysis*
> - *Content Creation*
>
> http://microsoftio.partnersalesresources.com/bpio.aspx

APO Maturity Levels

Let's now move our focus to the APO model. Before we explain what this really is, we will spend some time with the maturity levels Microsoft has identified for the assessment of the APO model. There are four optimization levels (see Figure 5-1).

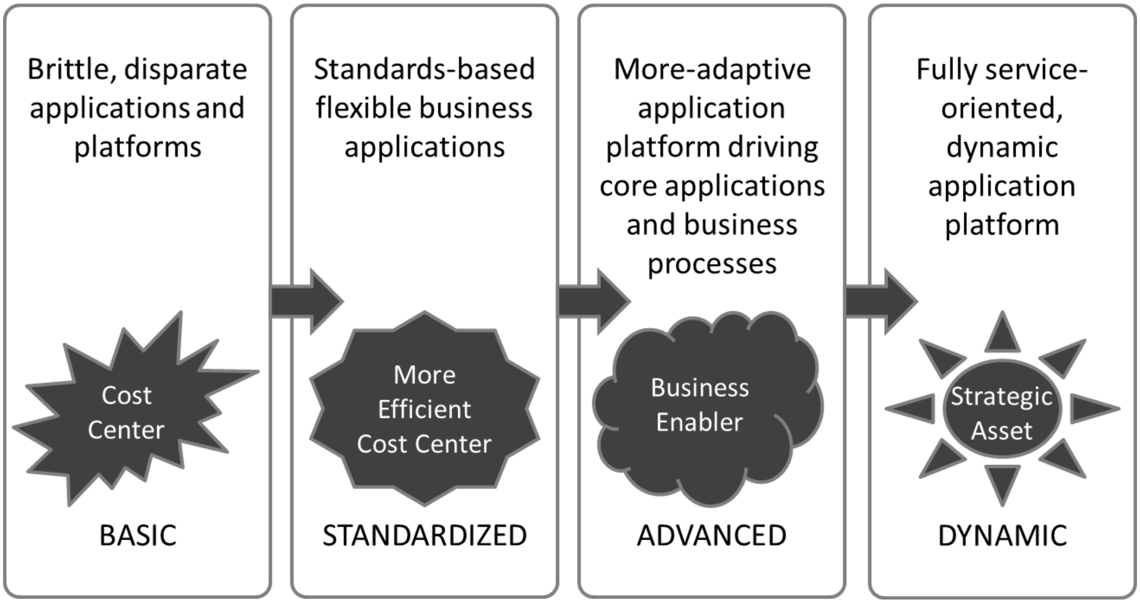

Brittle, disparate applications and platforms
Cost Center
BASIC
Standards-based flexible business applications
More Efficient Cost Center
STANDARDIZED
More-adaptive application platform driving core applications and business processes
Business Enabler
ADVANCED
Fully service-oriented, dynamic application platform
Strategic Asset
DYNAMIC

Figure 5-1*. The four optimization levels for categorizing an IT organization (as defined by Microsoft)*

Basic

When a company is classified as a *basic* organization, it is characterized by brittle, disconnected applications and platforms. This fact hinders rapid adjustments to business changes and also hinders the rapid development and interoperability of business-critical applications. The organization makes no real use of business processes, or these processes (if they exist) are often ill-defined. The processes are definitely not automated in any way. Such an organization probably has no tool for collaboration between teams and team members, and definitely lacks the clear connection between IT and business that is crucial for a company to have.

The development process is probably quite rigid, which makes development hard to control. All in all, this leads to higher costs, application backlogs, and lower IT productivity. The IT department is probably seen as just a cost to management and its true potential as a strategic asset is clouded by the problems.

Standardized

The standardized organization has begun to use industry standards broadly across departments, as well as with business partners. These standards could be as simple as starting to use XML, for instance. Furthermore, such an organization has started to take control of their development and data infrastructure, enabling the use of business intelligence reports and analytics. They have also started to automate some of their business processes. The IT department has slowly begun to be seen as a business enabler that could provide help in building more-adaptive systems quickly.

Advanced

At an advanced level, IT is truly seen as a business enabler and partner. Now infrastructure and systems are more easily managed throughout their lifecycles. The business processes are well-defined and well-known. The business side has begun to truly take advantage of IT and can rely on the IT department to quickly make adjustments when changes in the business occur. Such a company has standardized a robust and flexible application platform for the most critical systems and processes.

Dynamic

A dynamic organization is fully aware of the strategic value of its infrastructure. The organization knows IT can help run the business efficiently and stay ahead of market competitors. Costs are controlled to the company's maximum ability. All processes are automated and integrated into the technology, enabling IT to adjust to business changes and needs. The collaboration between the business and the IT department is working smoothly, as well. It is also possible to measure the effects of business benefits of IT investments, which is a strong argument when showing the value of IT. This kind of organization has also used SOA to the fullest so that cost-effective systems can be developed.

APO Capabilities

Today the software issues that held organizations back from moving up in this hierarchy are available and progress is being made, but surprisingly many remain at the Basic and Standardized levels. With this background, we are ready to have a look at the capabilities included in the APO model. Microsoft defines five capabilities, as shown in Figure 5-2.

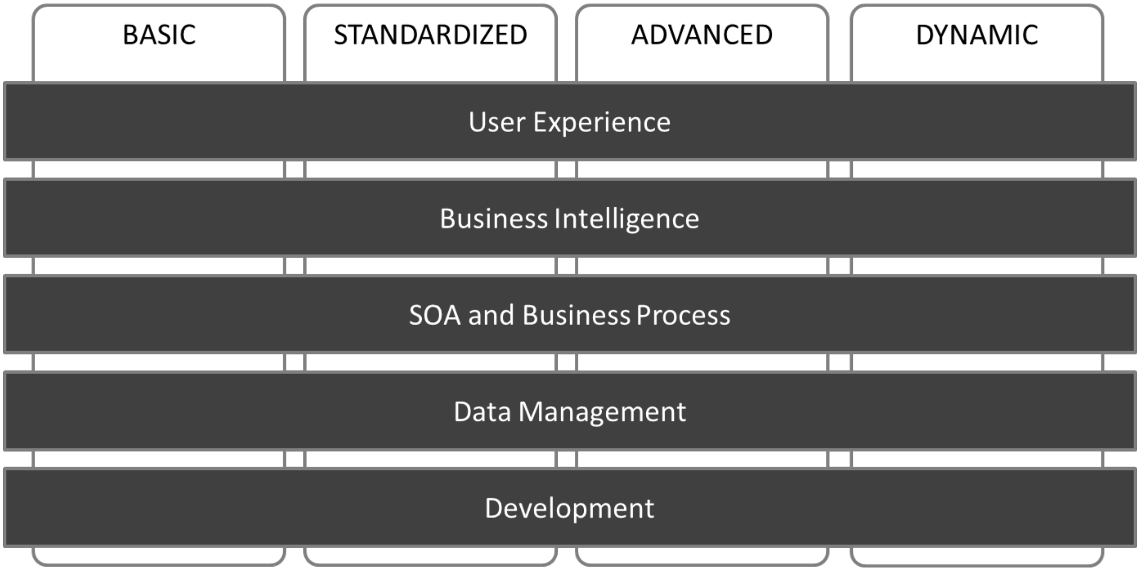


Figure 5-2. *Microsoft has defined five capabilities for their APO model (https://partner.microsoft.com/40029360).*

User Experience

User experience (UX) is an unappreciated area. UX is important, but most often this capability is not included in projects as a special field. We far too often rely on developers to solve UX problems without thinking of the effects bad UX design could have. Not many developers are skilled in this area and the importance and value of this field is included in this capability. Usability should be a higher priority in most cases. We have seen projects that were considered failures because the user experience was too technical. The developers had a technical view on how to work in the application (or system) and had designed it with this as viewpoint priority. The design was not in line with how the end user really worked, so the user interface needed a lot of rework, resulting in higher costs and a delayed project.

Business Intelligence

Microsoft also identifies business intelligence (BI) as a capability. Microsoft and many others have a vision that business insight should be provided to all employees. This leads to faster, more reliable, and more relevant decisions in the organization, because all members of the organization have access to the right information to support good decision making. We find areas such as data mining, reporting, data warehousing, data integration, analysis, and more here.

SOA and Business Process

SOA and business process is another capability. SOA can be a great thing to implement in our organizations, as you might have heard over the years. But SOA in our opinion has turned a little cold lately, and we don’t hear much about it. However, this capability focuses on the integration between Business Process Management (BPM) and SOA, and some experts contend that this integration should form an intimate link

Having effective business processes that we are able to quickly adjust to new or changed business needs is essential for an organization these days. We need ways to manage our processes and then automate them in our IT infrastructure. BPM will help with managing the processes, and SOA will help with implementing them in our IT environment.

Data Management

Data management covers what an organization should consider when integrating data management and analysis software. How is the data storage handled? Will it support the business-critical systems reliably? This capability also covers how database development is being carried out, how well the database team is integrated into the development projects, and so on. The main focus is to determine how best to build an integrated, well-managed, and always-connected data infrastructure to support our most demanding and mission-critical applications.

Development

Let's look at the development capability. Here we find the things that can enable an organization to develop applications that connect business processes to meet business needs. It covers areas such as what kind of development platform the organization uses, whether a development process is in place, how the development team and projects are organized, how visibility into the process of a development project is going, and so on.

Application Platform Capability Assessment

Next we will take a look at what an assessment can look like.

This assessment is called the Application Platform Capability Assessment and exists in three versions:

- Application Lifecycle Management
- Business Intelligence
- SOA and Business Processes

We will use the Application Lifecycle Management for our discussion because this one is most relevant for covering the development process and the ALM process. It covers all aspects of the ALM process and is very extensive. So in order to get good coverage on what parts of the ALM process you can improve, this is the assessment you should use.

When starting this assessment you can see that we have two options:

- Start Individual Assessment
- Start Team Assessment

We will use the Team Assessment. One big difference with this assessment compared to the Individual Assessment, aside from the number of questions and the detail level in them, is that it is intended to be filled out by more than one person. Microsoft also encourages the use of a partner when gathering information about your organization. The best thing about using a partner for such an assignment is that you get an independent view on the answers and the state of the organization.

The ALM Assessment for Teams includes many areas; it has eight practice areas, all divided further into a various number of practices. The assessment has about 200 questions (this figure is subject to change), so it covers a great deal of material.

The following list shows the eight practice areas and their practices:

- Architecture and Design
    - Architecture framework
    - Analysis and design
    - Database modeling
- Requirements Engineering and User Experience
    - Elicitation
    - Requirements analysis
    - Requirements management
    - Traceability
    - UX research
    - UI design and prototyping
    - UI implementation
    - End-user documentation

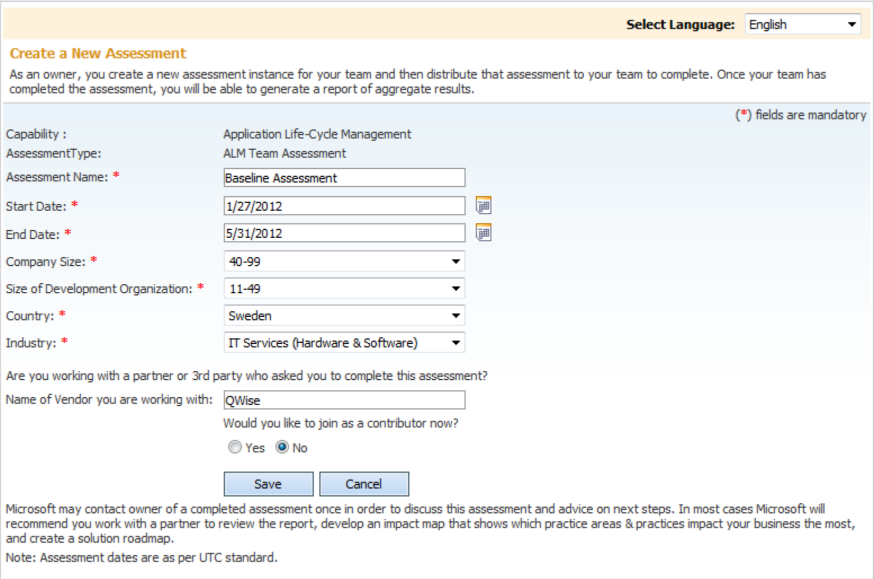
- Development
    - Code writing
    - Code analysis
    - Code reuse
    - Code reviews
    - Quality metrics
    - Database development
- Software Configuration Management
    - Collaborative development
    - Database change management
    - Version-control repository
    - Release management
    - Build management
    - Change management
- Governance
    - IT governance maturity
    - Application portfolio management
    - Compliance management
- Deployment and Operations
    - Designed for operations
    - Deployment
    - Environment management
    - Operations
    - Customer support
    - Database deployment
- Project Planning and Management
    - Project initiation
    - Project planning
    - Project monitoring and control
    - Risk management

- Stakeholder management
- Project close

- Testing and Quality Assurance
  - Test resource management
  - Test planning
  - Test management
  - Test types
  - Database testing

Starting the Assessment

When you start an assessment, you begin by filling in some information about the company. You can see in Figure 5-3 that you also set a time frame indicating the period that you'll allow people to add information into the assessment. You also can fill out the name of the partner you work with during the process.


Select Language: English

Create a New Assessment

As an owner, you create a new assessment instance for your team and then distribute that assessment to your team to complete. Once your team has completed the assessment, you will be able to generate a report of aggregate results.

(*) fields are mandatory

Capability :	Application Life-Cycle Management
AssessmentType:	ALM Team Assessment
Assessment Name: *	Baseline Assessment
Start Date: *	1/27/2012
End Date: *	5/31/2012
Company Size: *	40-99
Size of Development Organization: *	11-49
Country: *	Sweden
Industry: *	IT Services (Hardware & Software)

Are you working with a partner or 3rd party who asked you to complete this assessment?

Name of Vendor you are working with: QWise

Would you like to join as a contributor now?

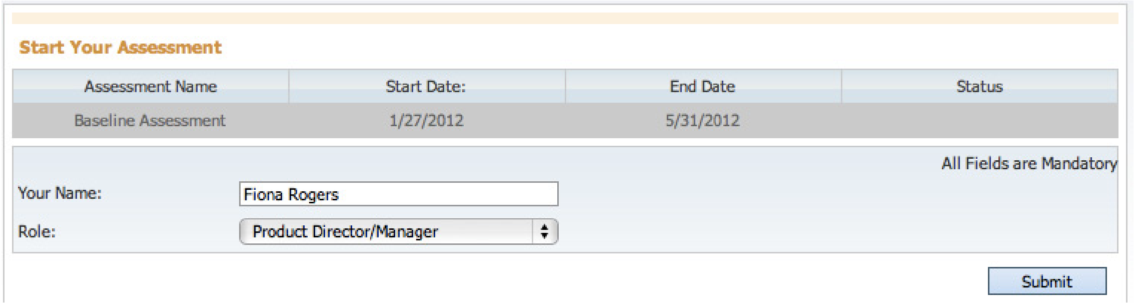
○ Yes ● No

Microsoft may contact owner of a completed assessment once in order to discuss this assessment and advice on next steps. In most cases Microsoft will recommend you work with a partner to review the report, develop an impact map that shows which practice areas & practices impact your business the most, and create a solution roadmap.

Note: Assessment dates are as per UTC standard.

Figure 5-3. *Starting an Application Platform Capability Assessment as an owner*

The creator, or owner, of the assessment sends an e-mail to all contributors containing the URL of the assessment. Once a participant opens the URL, they are welcomed with a page as seen in Figure 5-4.



Start Your Assessment

Assessment Name	Start Date:	End Date	Status
Baseline Assessment	1/27/2012	5/31/2012	

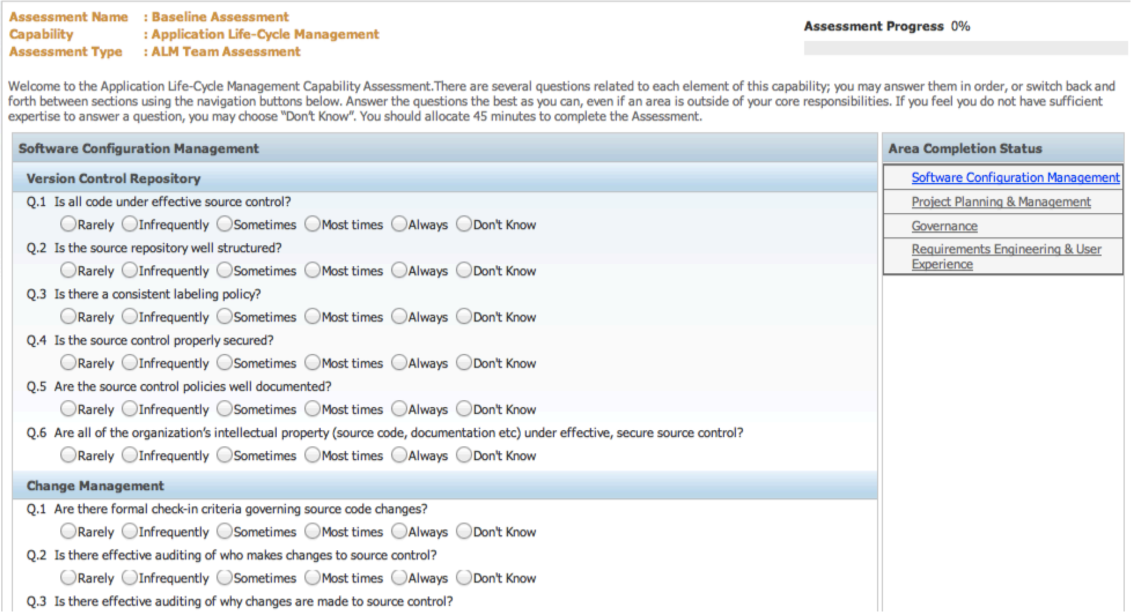
All Fields are Mandatory

Your Name: Fiona Rogers

Role: Product Director/Manager

Figure 5-4. Starting an Application Platform Capability Assessment as a contributor

When you enter the assessment, you'll have one page for each of the practice areas. As you know, these areas are divided into practices, and these are displayed as sections on each practice area page. Each section displays the questions for each practice (see Figure 5-5). Some practices have only one question, and others have more, so variance is great.



Assessment Name : Baseline Assessment
Capability : Application Life-Cycle Management
Assessment Type : ALM Team Assessment

Assessment Progress 0%

Welcome to the Application Life-Cycle Management Capability Assessment.There are several questions related to each element of this capability; you may answer them in order, or switch back and forth between sections using the navigation buttons below. Answer the questions the best as you can, even if an area is outside of your core responsibilities. If you feel you do not have sufficient expertise to answer a question, you may choose "Don't Know". You should allocate 45 minutes to complete the Assessment.

Software Configuration Management

Version Control Repository

Q.1 Is all code under effective source control?
○Rarely ○Infrequently ○Sometimes ○Most times ○Always ○Don't Know

Q.2 Is the source repository well structured?
○Rarely ○Infrequently ○Sometimes ○Most times ○Always ○Don't Know

Q.3 Is there a consistent labeling policy?
○Rarely ○Infrequently ○Sometimes ○Most times ○Always ○Don't Know

Q.4 Is the source control properly secured?
○Rarely ○Infrequently ○Sometimes ○Most times ○Always ○Don't Know

Q.5 Are the source control policies well documented?
○Rarely ○Infrequently ○Sometimes ○Most times ○Always ○Don't Know

Q.6 Are all of the organization's intellectual property (source code, documentation etc) under effective, secure source control?
○Rarely ○Infrequently ○Sometimes ○Most times ○Always ○Don't Know

Change Management

Q.1 Are there formal check-in criteria governing source code changes?
○Rarely ○Infrequently ○Sometimes ○Most times ○Always ○Don't Know

Q.2 Is there effective auditing of who makes changes to source control?
○Rarely ○Infrequently ○Sometimes ○Most times ○Always ○Don't Know

Q.3 Is there effective auditing of why changes are made to source control?

Area Completion Status

Software Configuration Management
Project Planning & Management
Governance
Requirements Engineering & User Experience

Figure 5-5. Answering an Application Platform Capability Assessment as a contributor

So what kinds of questions can you expect from this assessment? We will show some of them next so you can get an idea of how the practices are examined. Just as you would expect, the questions are more detailed in each area compared to the shorter APO assessment. Let's take a look at two examples.

Example 1: Assessing Requirements Handling

This first example question asks about how you handle requirements in the organization. It tries to find out whether you update the original requirements when changes happen.

Requirements Engineering and User Experience—Requirements Management practice
Q: Are requirements updated as requirements change?
A: 1. Rarely, 2. Infrequently, 3. Sometimes, 4. Most times, 5. Always, 6. Don't know

In many organizations, we have seen requirements remain fixed in the requirements specification no matter what happens to the requirements themselves. What usually happens is that the developers go ahead and change the functionality to reflect the requirement change (that might have come as an e-mail or by phone), without changing the documentation.

So discussing this question at the interview will tell you more than the question itself would. Try to find out whether the organization has a requirements system or application, and if they do, whether it is used. You can also ask about how their change requests are handled and whether they have a process for that. If they have such a process, a developer would not implement the change. Instead, the developer would redirect the person initiating the change to the correct step of the change request process—usually by sending a formal change request to the project manager or whomever is in charge of this.

Example 2: Assessing Code Analysis

The next question we will show you covers code analysis. Code analysis enables us to make sure that developers follow a set of rules indicating how code must be written. Code analysis can include everything from naming conventions to more-specific coding practices. This analysis can be automated.

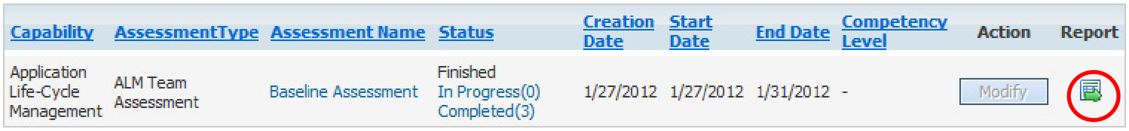
Development—Code Analysis practice
Q: Is there good static code analysis?
A: 1. Rarely, 2. Infrequently, 3. Sometimes, 4. Most times, 5. Always, 6. Don't know

There is no chance of having good code analysis without having it automated in some way. In Chapter 6, you will see that this is built into TFS, so we have access to it from there.

We use this question to find out more about the company's use of tools for automating the development process. It's a good starting point to dive into this subject, and to see whether they have other areas where automation is or is not used. The answers will help you better understand how the organization can benefit from implementing TFS (or any other ALM tool).

Viewing the Results

When all participants have answered their assessments, the assessment owner closes the assessment so that no one can enter any more information. To see the results, the owner then clicks the Generate Report button (see Figure 5-6).

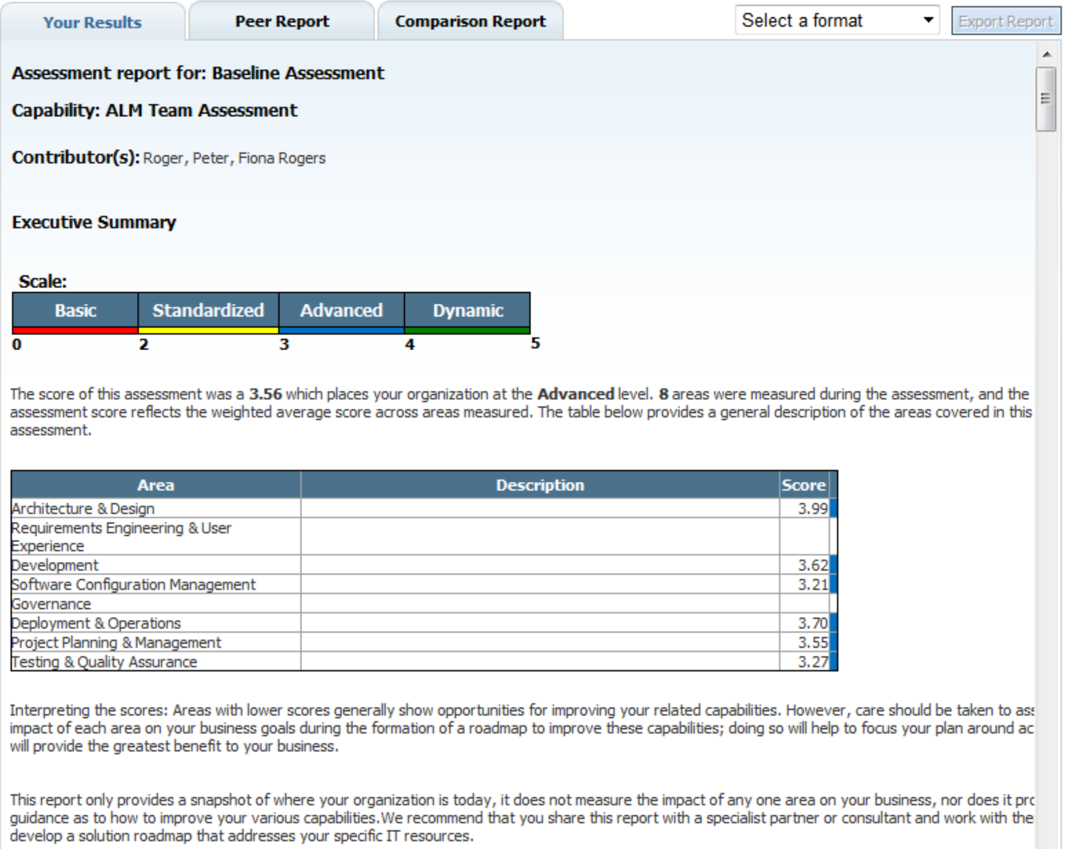


Capability	AssessmentType	Assessment Name	Status	Creation Date	Start Date	End Date	Competency Level	Action	Report
Application Life-Cycle Management	ALM Team Assessment	Baseline Assessment	Finished In Progress(0) Completed(3)	1/27/2012	1/27/2012	1/31/2012	-	Modify	

Figure 5-6. Generating a report for an Application Platform Capability Assessment

The questions are rated on a five-degree scale (1, 2, 3, 4, and 5) with a sixth choice being the answer "Don't know." The best score is 5, and the lowest is 1. (This is exactly like the scoring system we had in the Swedish schools a few years ago!)

The system calculates the medium score for each capability, for each capability area, and for the whole assessment and presents it graphically to the user (see Figure 5-7).



Assessment report for: Baseline Assessment

Capability: ALM Team Assessment

Contributor(s): Roger, Peter, Fiona Rogers

Executive Summary

The score of this assessment was a **3.56** which places your organization at the **Advanced** level. **8** areas were measured during the assessment, and the assessment score reflects the weighted average score across areas measured. The table below provides a general description of the areas covered in this assessment.

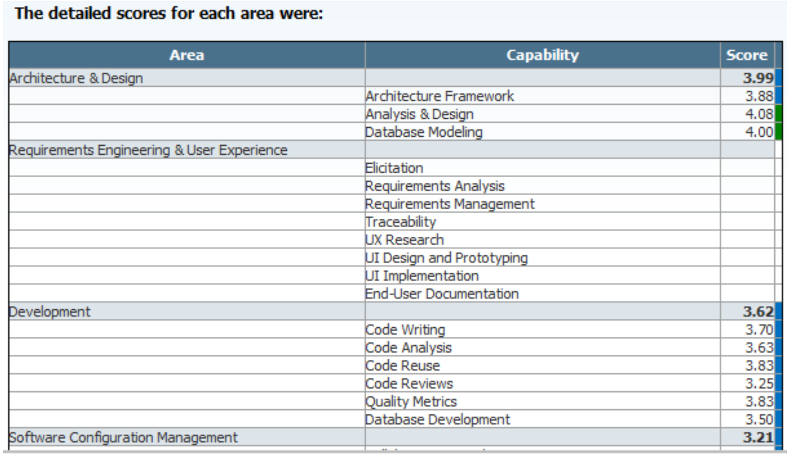
Area	Description	Score
Architecture & Design		3.99
Requirements Engineering & User Experience		
Development		3.62
Software Configuration Management		3.21
Governance		
Deployment & Operations		3.70
Project Planning & Management		3.55
Testing & Quality Assurance		3.27

Interpreting the scores: Areas with lower scores generally show opportunities for improving your related capabilities. However, care should be taken to ass impact of each area on your business goals during the formation of a roadmap to improve these capabilities; doing so will help to focus your plan around ac will provide the greatest benefit to your business.

This report only provides a snapshot of where your organization is today, it does not measure the impact of any one area on your business, nor does it prc guidance as to how to improve your various capabilities.We recommend that you share this report with a specialist partner or consultant and work with the develop a solution roadmap that addresses your specific IT resources.

Figure 5-7. The report for an Application Platform Capability Assessment

You will see a text overview of the whole assessment. Our demo here shows a pretty good score of 3.56, which puts this organization at the Advanced level. In the table below this score, you can see the individual practice area scores. You can see the maturity level for each area as well. This information is a pretty good summary for management to look at. But if you want to see more detailed information, you can scroll down the web page to see the score and maturity level for each practice, as seen in Figure 5-8.

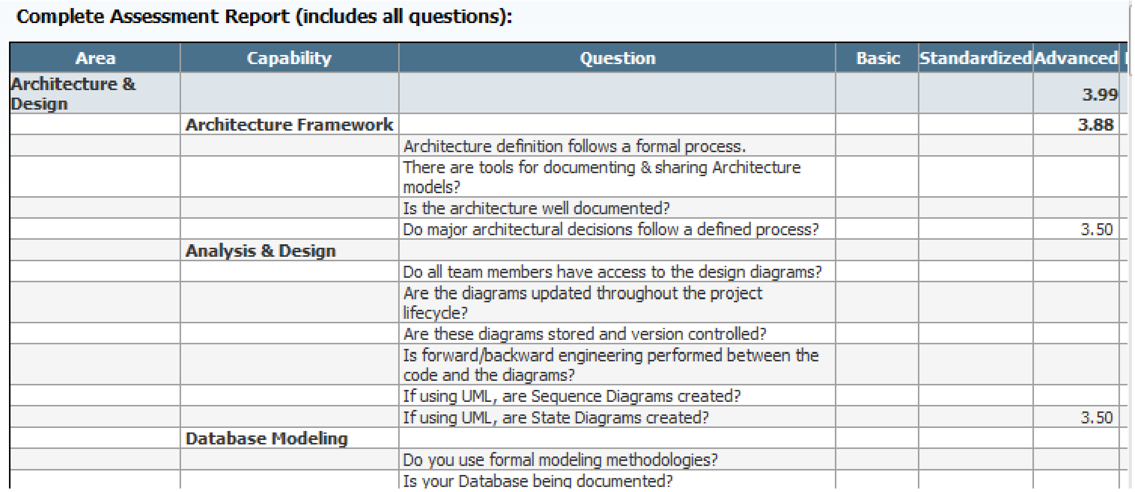


The detailed scores for each area were:

Area	Capability	Score
Architecture & Design		**3.99**
	Architecture Framework	3.88
	Analysis & Design	4.08
	Database Modeling	4.00
Requirements Engineering & User Experience		
	Elicitation	
	Requirements Analysis	
	Requirements Management	
	Traceability	
	UX Research	
	UI Design and Prototyping	
	UI Implementation	
	End-User Documentation	
Development		**3.62**
	Code Writing	3.70
	Code Analysis	3.63
	Code Reuse	3.83
	Code Reviews	3.25
	Quality Metrics	3.83
	Database Development	3.50
Software Configuration Management		**3.21**

Figure 5-8. Report detail from the score for each practice

Now you can pinpoint any problem practices, which are practices with lower scores. You can look for the color red or yellow in the right column (not shown in Figure 5-8) to quickly identify such a practice. A manager might want to dive even deeper into the results, so farther down you will find the score for each question (see Figure 5-9). This setup gives you the capability to identify exactly where you have problems, in which practice—and then to use this information for planning corrective measures.



Complete Assessment Report (includes all questions):

Area	Capability	Question	Basic	Standardized	Advanced
Architecture & Design					**3.99**
	Architecture Framework				**3.88**
		Architecture definition follows a formal process.			
		There are tools for documenting & sharing Architecture models?			
		Is the architecture well documented?			
		Do major architectural decisions follow a defined process?			3.50
	Analysis & Design				
		Do all team members have access to the design diagrams?			
		Are the diagrams updated throughout the project lifecycle?			
		Are these diagrams stored and version controlled?			
		Is forward/backward engineering performed between the code and the diagrams?			
		If using UML, are Sequence Diagrams created?			
		If using UML, are State Diagrams created?			3.50
	Database Modeling				
		Do you use formal modeling methodologies?			
		Is your Database being documented?			

Figure 5-9. You can see the score for each question in the assessment.

You probably want to be able to download this report so you can use it internally, and Microsoft allows you to do this. You can choose to save the report in different file formats so that you can process it any way you want.

■ **Note** You do not get any financial information in the Application Platform Capability Assessment report, only maturity scores at different levels of detail.

How to Use the Results

When we assess an organization's ALM process, we need to gather as much information as possible about the client and the client's organization. This is hard work if done manually. One would say it is close to impossible to collect that amount of information from so many people in the organization in a cost-effective way, without the use of a tool.

We used the questions from Microsoft's assessment tools as a basis for conducting interviews with people from ALM organizations. We gathered people from all aspects of the ALM process, making sure both the business side and the IT side were represented. Then we spent an hour to an hour and a half discussing these questions in one-on-one meetings. After conducting all interviews, we completed the assessment ourselves and used the interview result as a basis for answering the questions. In this way, we obtained a pretty realistic view of the organizations and their ALM processes. The results have also been better and have been more accepted by the organizations when we have done it this way as compared to when we have let only one person complete an assessment.

There are several ways to use the technique described in this chapter. We have tried it a few ways. Let's first start with a few comments on the assessments themselves. Tools are good in most cases. Tools can help us with many tasks and simplify our lives greatly, but tools for assessing a complete ALM process? Could that work?

Using the Application Platform Capability Assessment

The team assessment dives down deep. Keep in mind that a tool cannot elicit all the nuances that an observer can. We use the Application Platform Capability Assessment questions as a basis for interviews, and make sure that we interview people about their special fields only. Architects answer architect questions, project managers answer project management questions, and so on. This approach has worked very well.

We strongly recommend using an external partner for the assessments. It is often easier to look at an organization when there are no strings attached, and no manager to answer to.

Why Do an Assessment?

Why should you spend the time and money doing an ALM assessment? The best reason is that before implementing TFS (or any other ALM tool for that matter), you need to know what the potential pitfalls in the ALM process really are. Every process has room for improvement, and the assessment is a very good way of finding out where improvements are most needed. You need to have as clear a picture as possible of the organization's maturity level so that you can better anticipate what actions are needed for the organization to improve and thus be more effective.

The value of an assessment can be summarized in terms of what it provides. The assessment:

- Gives an analysis of the strengths and weaknesses with the current way of working
- Gives a foundation for prioritizing the ALM effort in the company
- Gives input to creating a roadmap for the improvement process
- Gives a baseline for follow-up assessments to measure the impact of an improvement project

An ALM Assessment will help your organization understand the current situation and make informed choices on the way to improve the ALM process. Often we think we know where the problem is, but before doing a proper analysis, it is hard to say. Making changes to the wrong thing(s) ultimately costs a lot of money—money better spent on correcting the real problem, which in the end could save money.

An ALM process is not something to implement all at once. You do it best little by little, piece by piece, starting with the lowest hanging fruit. If you just can show the decision makers the improvements of smaller actions, it becomes easier to get them to fund the rest as well.

Although the application of this assessment is limited in this chapter, you will receive a much fuller picture if you obtain the complete book, *Pro Application Lifecycle Management with Visual Studio*. We come back to this assessment throughout the book and tie together the topics with corresponding parts of the assessment.

Summary

This chapter has discussed the value of doing an assessment of the ALM process before implementing TFS (or any other ALM tool of your liking). The ALM assessments Microsoft offers are good but are best used in collaboration with an external partner carrying out the process in the form of interviews.

Use the assessment as a baseline for evaluating the impact of an improvement project. After a change in your process, you can perform the assessment again and again to measure and make sure you are moving in the right direction.

The next chapter concludes our coverage of ALM. Chapter 6 explains how Team Foundation Server (TFS) supports the basic pillars of ALM and the issues addressed by ALM 2.0+.

CHAPTER 6

ALM Using TFS

Adapted from Chapter 5 of Pro Application Lifecycle Management with Visual Studio, 2/e

In this chapter, we will discuss a tool which will make it clear why ALM is an important process for organizations engaged in IT development. A good implementation of ALM will help the organization deliver better business value to fulfill its business needs. Automating tasks by using tools such as Visual Studio and TFS can support this process.

In this chapter, you will learn how TFS can be used to fulfill the three main pillars of ALM and the issues addressed by ALM 2.0+, which we covered in Chapter 4. You will start with an overview of ALM and of TFS and then move on to the specifics of using TFS for ALM.

Application Lifecycle Management Overview

As you may recall from Chapter 4, there are three main pillars of an ALM process:

- *Traceability of relationships between artifacts.* The lack of traceability can be a major cost driver in any enterprise. There must be a way of tracing the requirements all the way to delivered code and back again—through architect models, design models, build scripts, unit tests, test cases, and so on. Practices such as test-driven development and configuration management can help, and these can be automated and supported by TFS.

- *Automation of high-level processes.* There are approval processes to control handoffs between analysis and design. There are other handoffs among build, deployment, testing, and so on. Much of this is done manually in many projects, and ALM stresses the importance of automating these tasks for a more effective and less time-consuming process.

- *Visibility into the progress of development efforts.* Many managers and stakeholders have limited visibility into the progress of development projects. Their visibility often comes from steering group meetings during which the project manager goes over the current situation. Other interest groups such as project members may also have limited visibility of the whole project even though they are part of it. This often occurs because reporting is hard to do and can involve a lot of manual work. Daily status reports can quite simply take too much time and effort to produce, for example, especially when we have information in many repositories.

Other important topics that ALM 2.0+ addresses are as follows:

- *Improving collaboration*. Collaboration is needed between teams, team members, stakeholders, and users, just to mention a few relationships. When development is spread around the world in different locations, collaboration can be hard to manage without the help of a proper tool.
- *Closing the gap between IT and business*. The big gap between IT and the business side of an organization is a serious problem for organizations, preventing us from delivering the greatest business value we can achieve in our projects.
- *Using one tool*. The complexity of using several tools for solving project issues as a team member can be tough and costly, as well. Switching between tools can be a cost driver. Using one tool enabling us to add plug-ins and use more features directly in our ordinary GUI instead of switching between applications is preferable. So if you have several roles in a project, you can still use one tool to get the job done.
- *Enhancing role switching*. ALM also addresses the potential to use one tool when switching among different roles in a project. In many cases, project members play several roles in projects. A developer, for instance, might also work with tests or databases. If that person can use the same GUI for all tasks, there will be minimal overhead for switching between these roles.

Team Foundation Server Overview

TFS has come a long way toward fulfilling the ALM vision, but it does not cover everything. TFS is an open and extensible product that will let us adjust its features to our needs and add the things it might lack at this point to support our specific needs. It is also important to know that Microsoft is spending a lot of time, energy, and money on developing this product further. It is not a toolset that will go away quickly it is one of the most important toolsets in the Microsoft ecosystem.

Team Foundation Server

You can see that the heart of ALM in the Visual Studio world is TFS, as shown in Figure 6-1.

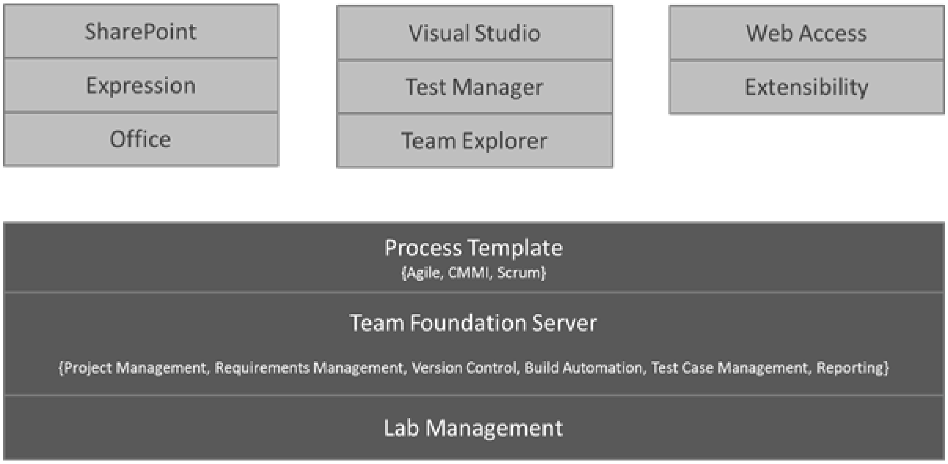
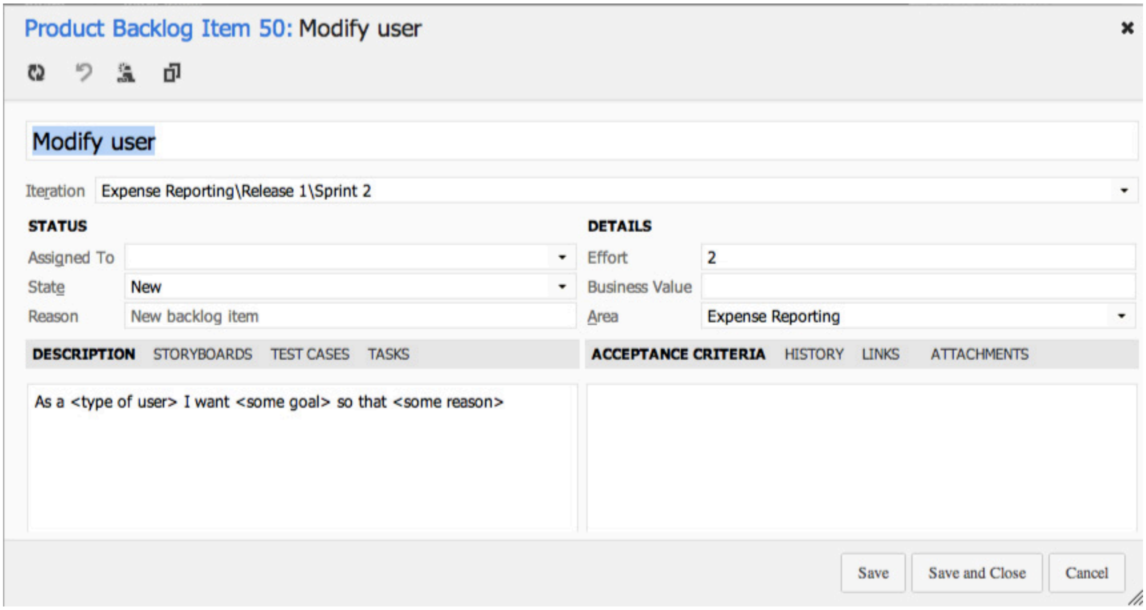


Figure 6-1. *Visual Studio Suite—an overview*

TFS exposes different functions and services for developers, project managers, version control, reporting, build and work item tracking (see Figure 6-2). You will soon take a look at all of these in more detail. Not shown in this picture is that TFS uses Microsoft SQL Server as its data repository.

■ **Note** Work items are used to manage different types of information in TFS. We have work items for requirements, bugs, general tasks, and so on. To put it simply, a work item is a piece of work that must be completed in a project. The work item tracking system is one of the core parts of TFS for our ALM process implementation.



Product Backlog Item 50: Modify user

Modify user

Iteration: Expense Reporting\Release 1\Sprint 2

STATUS

Assigned To:

State: New

Reason: New backlog item

DETAILS

Effort: 2

Business Value:

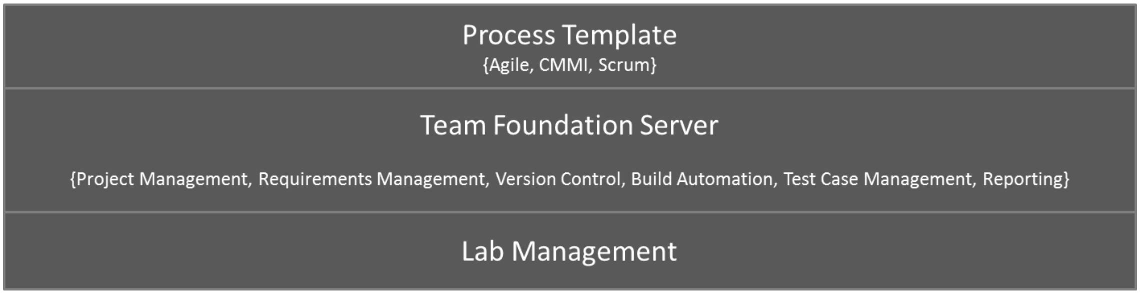
Area: Expense Reporting

As a <type of user> I want <some goal> so that <some reason>

Figure 6-2. *The heart of TFS—a work item. In this case, a product backlog work item.*

Process Template

What keeps all of these services together is the process template (see Figure 6-3). This is a very interesting part of TFS. The template helps us visualize and automate tasks and steps that the process includes. It helps us by providing document templates for requirements specs, test cases, scenarios, handoffs, and other artifacts we should produce.



Process Template
{Agile, CMMI, Scrum}

Team Foundation Server
{Project Management, Requirements Management, Version Control, Build Automation, Test Case Management, Reporting}

Lab Management

Figure 6-3. *The process template customizes TFS behavior*

Most companies use some kind of process for their development or ALM. Even though some companies don't think they have a process, they do. The process might not be written down, but the company still has ways of doing things that in reality is the process—for instance, naming conventions, where to store builds, how to handle change requests, and so on.

In many cases, we have seen companies with lots of money invested in their processes. They have sent staff to training, provided large process manuals, and so on. However, they have had problems getting project members to actually use the processes in their daily work. The excuses are many: the process is hard to understand, remembering all the process steps is difficult, the process is not automated or included in the tools, and many others.

The end result has been that project members use their own variant of the process, causing confusion during the project's lifetime. This also causes severe problems, as handoffs between the development team and the operations team are often difficult. A typical bad scenario can occur when a system has to wait for deployment because the infrastructure isn't in place for the new system. Operations was not involved (or perhaps even informed) during the project and suddenly they are expected to run the system on hardware they don’t have.

In TFS, we can implement our development process as a template that will be mandatory for all new projects. When we create a new project, we also create a new instance of the process template. And we don't have to stop at the development project level either. We can implement most parts of our ALM cycle in the template as well, enabling us to take advantage of TFS all along the way. The template helps us visualize and automate tasks and steps that the process includes. It helps us by providing document templates for requirements specs, test cases, scenarios, handoffs, and other artifacts we should produce.

The template also provides information about which reports we have available for each new project—reports that we use to retrieve information about the status of our projects and many other things. The template also contains information about one of the most important core parts of TFS: the work items. These can be adjusted as needed so we can make sure they contain the information the organization must have included with them. This information could be status information for a bug, for instance, such as Active, Resolved, or Closed.

This template is so flexible that we can develop and implement our own process, we can choose to use any of the three that Microsoft supplies, we can use a third-party template, or we can choose to customize the Microsoft templates to our own liking. We can also have several process templates in TFS so we can use different templates for different projects. Because TFS really is not used to its full potential without the process templates, we cannot stress enough that you should consider which templates you want to use and the information you want them to include.

Visual Studio Editions

Most developers will use Visual Studio to access the features of TFS. There are several editions available at the time of this writing:

- *Visual Studio Professional*, which gives the developer basic functionality, is perfect for the single developer or a very small team not using TFS. It has no CAL for TFS, but you can purchase one and get access to version control, work items, and so on.
- *Visual Studio Premium* is a more advanced version with a TFS CAL included. This version gives access to the most useful features of TFS like testing tools, database development tools, debugging and diagnostics, and more.
- *Visual Studio Ultimate* gives access to all features of TFS. Here we can get going with lab management, architecture tools, and more testing tools just to mention some of the features.

- *Visual Studio Test Professional* is the tool for testers. These tools are also included in the Ultimate edition but it lacks the development tools included in the other editions.
- *Team Explorer Everywhere* lets developers on other platforms like Eclipse on Mac have access to TFS. We will show you more of this later in the book. This is the perfect add-on for teams with development on multiple platforms like .NET and Java.

Web Access

All projects in TFS have their own web sites available. By using Windows SharePoint Services or Server, a team project portal is created when the project itself is created. By using this portal, or the more advanced Team System Web Access, we can access most of the functionality in TFS. The project portal lets us access the parts of TFS that are available from inside Visual Studio, from an easy-to-use interface, especially for nontechnical project members. Figure 6-4 shows what Web Access looks like.

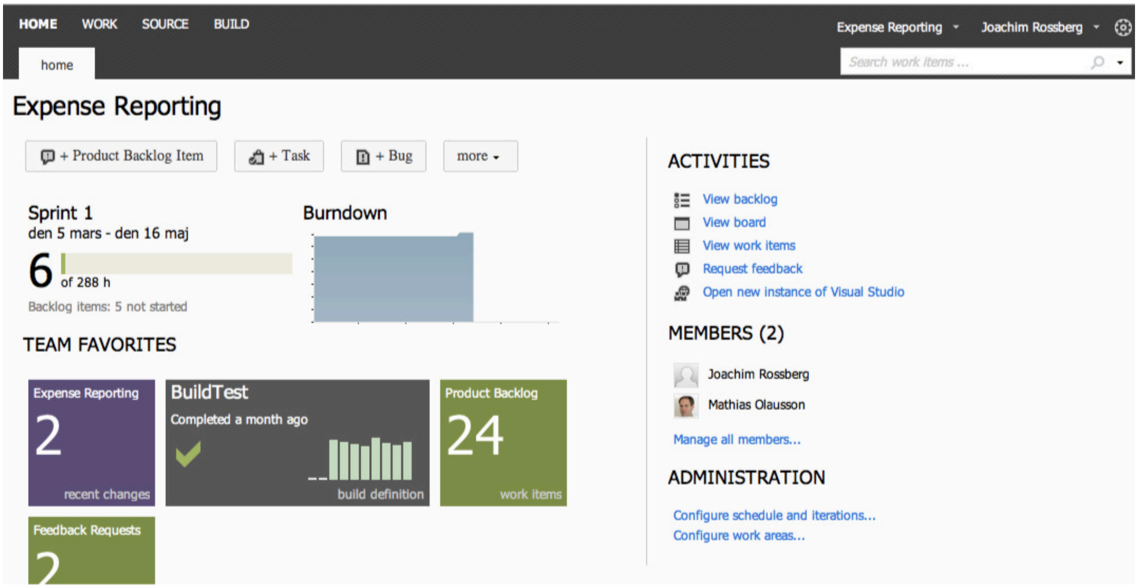


Figure 6-4. *The Team System Web Access start page on TFS*

Many of our customers use a team project portal primarily to provide access to reports and documents for nontechnical people not used to the Visual Studio interface. When we want to give an external user (such as a customer or remote stakeholder) access to work item creation and editing, or another more advanced task, we usually use Web Access.

Microsoft Office

Microsoft Office can be used by project managers, product owners or scrum masters, for example, wishing to use tools that are familiar to them, such as Microsoft Project and Microsoft Office Excel, during a project. The integration is very nice and valuable to these roles.

IDE Integration

When it comes to add-ins, one thing we should mention in particular is the Team Explorer. This tool can be used as an add-in to Visual Studio, and it gives access to TFS directly from within Visual Studio. From here you can open reports, add new work items, and run queries against the TFS database.

TFS is a flexible tool, as we have mentioned. It is also very extensible as all functionality can be accessed via web services. This is a very nice feature that enables us to build our own support for TFS in other applications, as well. Many third-party vendors have done this, and a wide array of add-ins and tools are available. My favorites came from Teamprise, a company that has built add-ins to Eclipse so that we can use TFS features in our Java development environment, as well. Teamprise was purchased by Microsoft, and its suite of client applications has been available as Team Explore Everywhere since TFS 2010. We do not cover Team Explorer Everywhere in this chapter, but briefly it offers the same IDE integration into both Eclipse and Visual Studio, allowing us to truly work as one team no matter whether you use Eclipse or Visual Studio. In the complete book, *Pro Application Lifecycle Management with Visual Studio*, we cover this solution in more detail.

Traceability

Having traceability in our ALM processes is key to the successful delivery and maintenance of our applications and systems. In Chapter 5, we discussed a company that stopped making changes to its systems just because no one ever knew where a change (or bug fix) might have its impact. We don't have to have such a situation.

These TFS features can help us with traceability so we can avoid such problems:

- Work item tracking
- Test-driven development/unit testing
- Automated builds/continuous integration
- Check-in policies
- Version-control system

Let's look at some of the specifics involved with these features, starting with how the work item tracking system implements traceability.

The TFS Work Item Tracking System

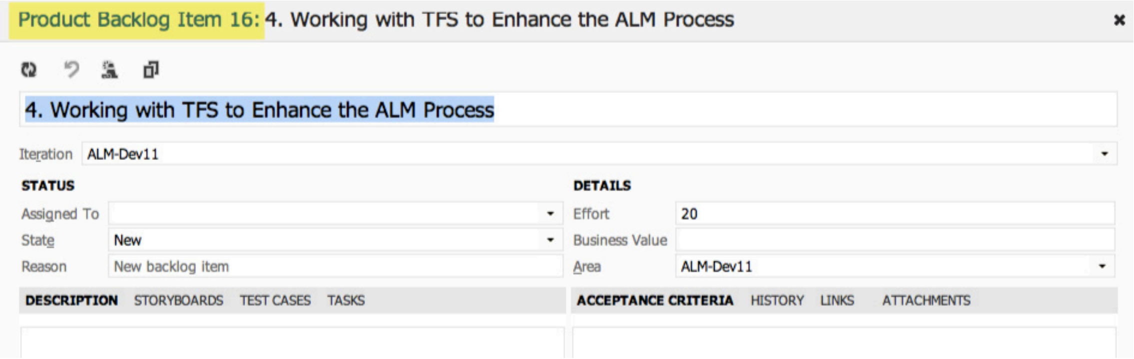
Sometimes it seems like we have tons of Post-its on our monitors and desks—each one containing at least one task we are supposed to take care of. We would like to track them in a tool that could help us, but often it just isn't possible. It could be that some tasks are connected with one project, others with another. We have tried writing them all down in an Excel spreadsheet and saving that to the computer.

But soon we find that this spreadsheet is located on our laptops, our customer's computer, our desktops, another customer computer, and so on. And we have no idea which one is the current version.

The same thing often occurs in projects. Project managers have their to-do lists for a project, and they all have their own way of keeping them updated. Let's say a PM uses Excel to keep track of the tasks—the status of tasks, whom they are assigned to, and so on. How can the PM keep the team updated with the latest to-do list? If the PM chooses to e-mail it, chances are that some won't save the new version to disk or will just miss it in the endless stream of e-mail coming into their mailboxes. Soon there are various versions floating around, and things are generally a mess.

In TFS, we have a task-tracking system at our service. The core of this system is represented by the tasks themselves, which as we said earlier are called *work items*. A work item can be pretty much what we want it to be. It can be a bug, a requirement of some sort, a general to-do item, and so on. Each work item has a unique ID that helps us keep track of the places it is referenced (see Figure 6-5). The ID lets us follow a work item, let's say a requirement, from its creation to its implementation as a piece of executable software (component).

Work items provide a great way for us to simplify our task management in a project while at the same time enabling traceability. No more confusion as to which version of the task list is the current one. No more manual labor for gathering status reports on work progress that are used only at steering group meetings. Now we have a solution that lets us collaborate more easily with our teams and enables all members and stakeholders to view status reports whenever they want. We can also more easily collaborate with people outside the project group by adding work items via the web client.



Product Backlog Item 16: 4. Working with TFS to Enhance the ALM Process

4. Working with TFS to Enhance the ALM Process

Iteration: ALM-Dev11

STATUS

Assigned To:

State: New

Reason: New backlog item

DETAILS

Effort: 20

Business Value:

Area: ALM-Dev11

Figure 6-5. *Each work item has a unique ID.*

TFS is so flexible in this regard that it lets us tailor the work items as we want them to be. The work item tracking system is one of the core components of TFS. This system enables us to create work items, or units of work, and can be used to enable traceability. We can use the work items included with TFS from the beginning, or we can choose to adjust these to our needs, or even create our own work item types. Each work item instance has a unique ID (as you saw earlier in Figure 6-8) that we can attach to the things we do in TFS. This enables us to follow one work item—let's say a requirement, for example—from its creation to its implementation as a piece of executable software (component). We can also associate one work item with others and build a hierarchy of work items.

The work items can contain information in different fields that define the data to be stored in the work item. This means that each field will have a name and a data type.

All work items can have different information attached to them. We can have information about to whom the work item is assigned and the status of the work at the moment (for example, a bug could be open, closed, under investigation, resolved, and so on). The State field can be modified so that each work item type can have its own state mechanism. This is logical because a bug probably goes through other states than those a general task goes through, for instance. We can also attach documents to the work item and link one work item to other work items. We can even create a hierarchy of work items if we want. Let's say that we implement a requirement as a work item and this requirement contains many smaller tasks. Then we can have the requirement itself at the top and nest the other requirements below that so we know which work items belong to which requirement.

When a bug is discovered, for instance, we can quickly follow the original requirement by its work item ID and see in which places in the code we might have to make some fixes. We can also see the associated work items so that we can evaluate whether other parts of the code need to be changed as a result of this bug fix.

TFS saves information about the work item on the data tier, which helps us follow the change history of the work item. We can see who created it, who resolved it, who closed it, and so on. The information in the databases can be used for display on reports, allowing us to tailor these depending on our needs. One report could show the status of all bugs, for instance. Stakeholders can see how many open bugs exist, how many are resolved, and much, much more. It is completely up to us how we choose to use the work items.

If we implement a requirement as a work item, we can use the work item ID to track this requirement through source code and to the final build of the executable system. By requiring all developers to add one or more work item IDs to the check-in using a check-in policy, we can enable this traceability.

Configuration Management Using TFS

In any (development) organization, we need to have control of the versions of our systems we have in production. If we don't have that, the overall ALM process will suffer, because we will suddenly lose traceability. This will make it harder to implement changes and bug fixes, because we won't know which versions we need to update.

Without the help of a proper tool, we soon will get lost in the variety of applications we have. TFS can help us with this in many ways. After a brief description of software configuration management, I will cover three of the most important concepts that have great support in TFS and Visual Studio tools:

- Version control
- Release management
- Build management

■ **Note** In software engineering, software configuration management (SCM) is the task of tracking and controlling changes in the software. Configuration management practices include revision control and the establishment of baselines, and are very important. There are several goals of SCM, including the following:

Configuration identification: Ensuring that we know what code we are we working with
Configuration control: Controlling the release of a product and its changes (version control)
Build management: Managing the process and tools used for builds

Defect tracking: Making sure every defect has traceability back to the source

If these issues are not covered by our ALM process, we could very soon find ourselves in a troublesome situation. It is crucial for the development teams to have full control over which versions of the applications exist, which are in production, and where. This topic is closely related to the portfolio management team, and generally a big company has one or more persons devoted to keeping track of this.

Version Control and Release Management in TFS

Using the version-control system in TFS, we can manage and control multiple revisions of the same information in our projects. This information can be source code, documents, work items, and other important information that we want to add version control to. When we want to work on an item under source control, we check it out to our local computer so we can start working on it. When work is done and tested, we check in our changes so the version on the server is updated.

The version-control features of Visual Studio Team System are powerful. They are fully integrated into the GUI, which is something that ALM 2.0+ prescribes as well. If you want to, you can access some of the features from project portal as well. Many people want to use the command line for their work, and TFS enables them to use the command line for working with version control as well.

However, if you do want to use Visual Studio to access the TFS version-control system, you can do that. The extensibility of TFS makes this possible. One example of this is the Team Explorer Everywhere suite of client applications that can access TFS, including the version-control system. You still will use the TFS repository, and have the ability to get reports and other information directly from TFS.

Build Management

A build is basically the process of taking the source code and all other items necessary in an application and building it into executable software. Team Foundation Build is the build engine in TFS and executes the build process as defined by the TFS settings. Team Foundation Build is built on the Microsoft build engine (MSBuild), which is the build platform for Microsoft and Visual Studio. You can attach unit tests to a build process so that you automatically run these every time the build process kicks off. Team Foundation Build is fully integrated into the Visual Studio GUI so you don't have to use separate tools for handling these tasks.

Team Foundation Build supports several types of builds:

- *Full builds:* We build everything in our project. This can be resource- and time-consuming.
- *Partial builds:* We build only one or more parts of the system.
- *Nightly builds:* Many projects build the system during nighttime. Especially if the build process takes a long time, this might be very handy.
- *Custom builds:* We can use the extensibility of TFS to create our own build types, or edit any of the existing ones.
- *Incremental builds:* We build only the components that have been changed since the last build.

You can also add a number of tasks that you want to be executed when running the build:

- Get the latest source code from the version-control system
- Compile sources
- Perform Architectural validation
- Run static analysis tool
- Execute unit tests
- Update work items
- Perform code coverage
- Calculate code churn (how many rows of code have been modified or added since last count)
- Produce build reports
- Drop exe/output into predefined location

Automation of High-Level Processes

Without one or more templates, TFS will not be used to its full potential, as you saw earlier in this chapter. You could still use its version-control system and some other tools, but the real value comes from using it to automate your ALM process. In the process template, your whole ALM process is defined.

The template defines the following:

- *Work items types:* Which work item types are necessary and what information they should have attached to them. You can also define the workflow for a work item. For a bug, you might have different states the item flows through, such as Active, Resolved, Closed, and so on.
- *Project phases:* By using areas and iterations, you can define the initial project phase setup of your projects. If you use RUP, you can define the process phases in that model, or you can create the first sprints of a scrum project. Areas and iterations are flexible, and you can create your own way of working through these concepts.
- *Document structure and templates:* The number of documents that should be produced during a project will differ depending on your process model. In the process template, you define the document structure you need and the templates you should use. For instance, you can add templates for requirements specifications or acceptance testing here.
- *Reports and queries:* In the process template, you can specify which reports and work item queries you need to have as defaults in your projects. You probably want reports and queries showing the progress of your project, such as the status of bugs or work remaining. You can create your own reports by using SQL Server Reporting Services or Excel, and add them to all projects by adjusting the process template.

- *Security:* The template also adds information about which users or user groups have access to what information. You can connect TFS groups to your Active Directory accounts, for instance.

The process template is the overall process for our ALM implementation. Many of our customers create different templates for different kinds of projects. They also create templates for operations, so that when a development project is finished and deployed, the operations staff can use their template to run the system until the system dies. A few customers have started creating a template for ITIL, for instance, and we are looking forward to seeing the result of that work.

It is important to remember that you can adjust the process to your needs. You should consider changing the default templates or even replacing them, rather than adjusting your own way of working to the templates that come with TFS out of the box. Microsoft enables this flexibility by letting you easily access the process templates to adjust them or to add new templates.

Visibility

Information about project status is important to all participants of a project—and we don't mean team members only, but stakeholders and decision makers, as well. As a project manager, we have spent too much time chasing down information to answer questions about the status of projects, how much work remains, and what the latest bug status is.

TFS provides two primary ways of enabling visibility:

- *Reports:* Reports are created by using SQL Server Reporting Services and accessing the TFS data tier directly. You can define and adjust these as you want. You can also use Excel to create reports if you prefer. Reports can show various kinds of information. (Covering them in detail is outside the scope of this chapter, but you will learn about them in detail if you obtain the complete book.)

- *Queries:* Queries are used to ask questions of the work item tracking service. One question might be how many bug work items you have. How many and which are dedicated to me? How many bugs are there? And so on. You can create new queries when necessary.

By using these two components, it will be easier to gather the information you need for your status reports for a steering group meeting or project meeting. You won't have to look around in several places and in several applications for this information anymore, but instead can use the automated reports and queries from inside TFS.

Project owners, project managers, and scrum masters will certainly benefit from TFS. Because TFS has all data in the same repository, you can more easily retrieve the correct information when you want it. The flexibility of the SQL Server database that stores all information is great. You can work with the data warehouse information just as you would with any other database.

By using the project portal or Web Access (see Figure 6-6), you can publish information (in the form of custom-built controls that we as users cannot change at this time) so that everybody who has privileges can see them. This is an easy way to make sure that information is available all the time. Just this little, relatively nontechnical improvement will off-load work from the project manager, freeing some of the PM's or PO’s time for better things.

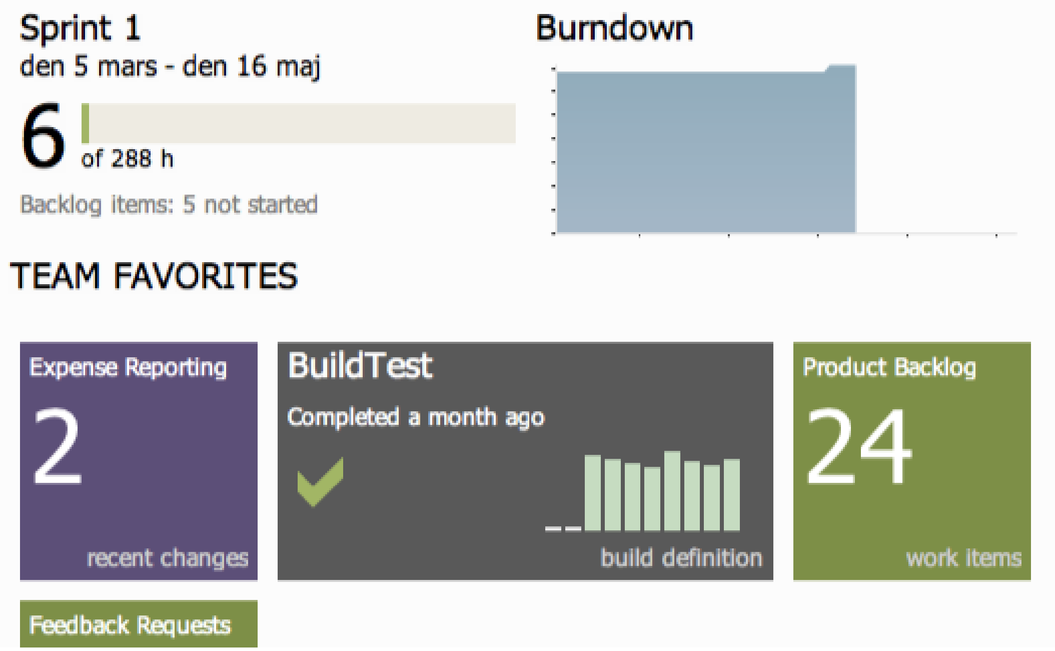


Figure 6-6. *Viewing reports from Team System Web Access*

Collaboration

As you know, TFS comes with Team Explorer, which is an add-in to Visual Studio. With this tool, the developer can access every aspect of a TFS project. The developer can view reports and queries, for instance, as well as access the document in the project. The developer can access the version-control system, build system, tests, and so on as well.

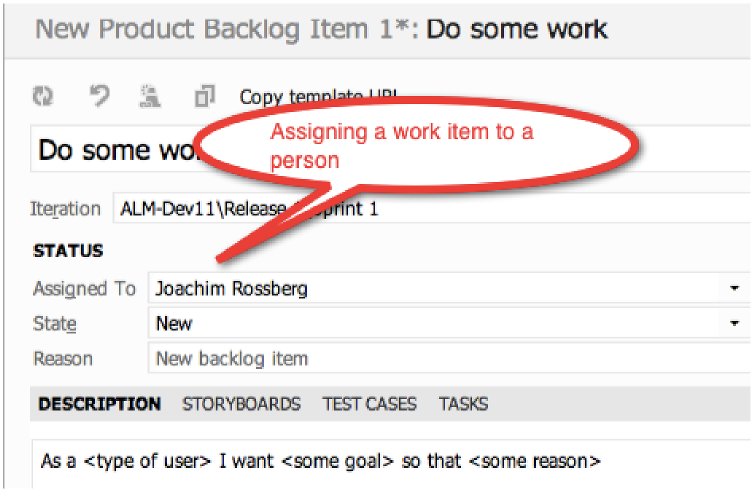
The Team Explorer is full featured but is still a tool for people used to working in Visual Studio. For us that is no problem, but for most project managers and stakeholders, the GUI is confusing. They want to have an easier-to-use tool to access the relevant information.

Each project that is created with TFS has a project portal created, as well. This portal gives us access to reports, documents, project process guidance, and other project-related information through a web interface. This enables people who are not used to the Visual Studio interface to easily retrieve the information they need.

Collaboration, of course, does not only mean giving access to information, even though this is as important as any other means of collaboration. Collaboration also means that we should be able to work together to fulfill one or more goals.

Work Items for Collaboration

You can use the work item features of TFS to enable your process workflows. Let's say a project manager, or anybody responsible for inputting requirements as work items into TFS, creates a new work item of the Scenario type. This scenario should probably be assigned to a developer to implement. The project manager uses the work item system to assign (see Figure 6-7) the scenario to a specific developer, in this case Joachim. Joachim continues to work on the scenario until it is ready for testing. He then assigns the work item to a tester who performs the testing. When the testing is done, the work item is perhaps closed. If a bug is found, either the tester or anyone finding the bug can use the work item tracking system to see who developed the scenario implementation and reassign it to that developer, in this case, Joachim again. TFS keeps track of who has worked on the work item so that you don't have to manually keep track of this.


New Product Backlog Item 1*: Do some work

Do some wo

Iteration ALM-Dev11\Release print 1

STATUS

Assigned To: Joachim Rossberg

State: New

Reason: New backlog item

As a <type of user> I want <some goal> so that <some reason>

Assigning a work item to a person

Figure 6-7. *Assigning work items to a specific person*

The Gap Between IT and Business

Closing the gap between IT and business is obviously a very tough problem to solve. TFS won't get us all the way, that's for sure. We don't think any tool ever will because so much depends on the people in the organizations, which is an important consideration. But tools can help us bridge the gap, so you should carefully consider how you can use them for this. We need to improve on our ALM process and way of working to start solving this. When we have a new way of working, TFS can support much of our efforts using, for instance, the process template to implement this new way of working.

The gap between IT and the business sides is often a question of work process. It requires considering many things, and when you have a solution or start working toward a solution, you must evaluate what parts of this work process you can automate and use tools for solving. One thing worth mentioning here is the use of the TFS Project Server Connector with TFS lets you integrate TFS with Microsoft Office Project Server. Having this integration will allow you to better control your resources and better automate this process as well. This way, you can align your portfolio management process better so that you can choose which things to work on more effectively.

Office/MS Project Integration

When we have run projects in the past, we have mostly used Microsoft Office Project to handle project planning, especially the Gantt diagram. We suspect this is the case for many of our fellow project managers, as well. In many cases, we have used this product not primarily because of the tool itself but because so many of our customers use Microsoft Office that it becomes natural for them to also use Project. Project has its strengths and weaknesses, as all tools do, and requires a good knowledge of the product to use it successfully.

Excel and Project are two tools that most companies use on both the business and the IT sides of the company. By being able to use these tools, business people can more easily be a part of the ALM process, because they can use a tool they are already used to working with. A nice feature here is that the communication between Office and TFS is two-way. This means that an update in TFS will be reflected in Office and the other way around. This allows for a dynamic way of working with TFS information.

Use of One Tool/Role Based

A good ALM tool should enable you to use add-ins that will provide new features inside one interface. If a developer needs testing features, you should be able to integrate them into the development tool. The developer should not have to switch tools to do testing tasks. This is also what Visual Studio offers. There is no context switching as team members can use the same GUI no matter what role they are performing at the moment. TFS is also extensible and lets you create your own add-ins as well as purchase third-party add-ins that will be accessible from inside of TFS.

Extensibility

When the built-in features of TFS are not enough, you can use the extensibility features to expand and enhance it. TFS is often seen as a closed black box that Microsoft ships, when it's more like an enterprise resource planning (ERP) system for ALM. Any ALM environment must be customized for an organization's processes and the existing applications and services.

Many of our customers have been a bit reluctant to customize TFS. They have instead tried to squeeze their way of working into the templates Microsoft provides with TFS. We think this is the wrong way to do it. Our suggestion is that you start the other way around. Start by asking yourself how your organization wants to work. This process involves all parts of the organization, from the business side to operations. Try to find agreement on how to work in the ALM process. By doing so, you will see that this also is a good start for collaboration in the company.

For instance, consider the work items and the information in them. If the fields and information in the MSF templates are not enough, extend or edit them. TFS lets us do this by changing the process template. You can choose to add the information that you need, and it will be stored in the TFS databases, so you can have access to it from within your reports and queries. Don't forget to change the reports or queries, as well; otherwise, you will not see your information.

Some of our customers have changed the workflow of a work item by adding more states to it, when the ones supplied have not been enough. Often we have used the TFS Power Tools to do this.

When you have an initial idea of how you want to conduct the ALM process, start looking into what TFS gives you out of the box. Use what can be used, change other things, and build your own solution where needed.

One great strength of TFS is its extensibility and flexibility. You can adjust the whole tool to fit most parts of your ALM process. If you want to, you can develop your own add-ins by giving support to roles not included from the start. We strongly encourage you to use these extensibility features, but in the end it is your choice.

Extensibility is a great way to integrate existing systems and potentially migrate some of them into TFS in order to reduce the toolset in the organization.

Summary

In our opinion, Team Foundation Server can help us implement a good, automated, and robust ALM process. There are features for all aspects of ALM 2.0+. Correctly used, these features will help us improve our ALM process, which in the end will give us better business value and more-successful projects.

The three pillars of ALM—traceability, process automation, and visibility—are all important for any organization to have. TFS is a great foundation on which to build our ALM solutions. TFS has work item tracking for traceability, process template implementation in the tool itself for process automation, and reports and queries for visibility. Through a project portal, accessible via the Internet, you can improve collaboration between all parties having an interest in your projects.

TFS is role based in the sense that it supports different development roles. It has support for architects, developers, DBAs, testers, and more. They are not separate tools either, but they are all accessible from a unified GUI. You can also add custom add-ins to the GUI and do not have to use several tools to get the job done.

Product owners and project managers have the capability to use tools they are already familiar with. Most use Excel or Project for project planning, and there is integration between these tools and TFS. We can easily sync information between these tools.

The extensibility of TFS makes it fairly easy to write your own code integrating TFS with other applications. This is an incredible strength of TFS, and something we should give Microsoft credit for.

So all in all, Team Foundation Server is a great foundation on which to build our Application Lifecycle Management process.

CHAPTER 7

Windows Azure Platform Overview

Adapted from Chapter 1 of *Windows Azure Platform, 2/e*

In recent years, cloud computing has emerged as a disruptive force in the information technology (IT) industry. Its impact is of the same magnitude as offshoring or virtualization. Gartner Research has identified cloud computing as one of the "top 10 disruptive technologies 2008–2012." According to Gartner, a *disruptive technology* is one that causes major change in the accepted way of doing things. For developers, architects, and operations, cloud computing has caused a major shift in the way of architecting, developing, deploying, and maintaining software services.

Cloud computing democratizes IT, similar to how the Internet democratized the consumer industry. The Internet opened up a vast ocean of accessible resources to consumers, ranging from free advertising-based searching to online banking. Cloud computing is bringing similar trends to businesses small and big. Businesses can now reap the benefits of agility by simply deploying their software in someone else's data center for a consumption fee. Hardware costs are out of the equation because of cloud service providers. These may sound like the hosting companies you already host your web sites on, but the big difference is that this is now a utility model built on highly scalable data center platforms. The cloud-computing wave is powerful enough for a company like Microsoft to start disrupting its own business model to invest in the opportunity.

In this chapter, I will cover some of the basics of cloud services and then jump into an overview of the Microsoft's Windows Azure platform. The second chapter in this section will provide a detailed discussion of the Windows Azure Compute architecture and service management components.

Introducing Cloud Services

As an introduction to our discussion, consider a typical scenario in today's medium to large enterprises. Assume a business unit has an immediate need to deploy a highly interactive niche web application (a micro-site) for a new product that will be released in five months. The application will provide the consumers a detailed view of the product and also the ability to customize and order the product right from the web site. The business unit has the budget but not the time and resources to implement it, and this deployment needs to happen in the next three months for it to be ready for the launch.

The IT hosting team understands the requirement, but to deploy an application with IT resources requires coordination among hardware, software, operations, and support teams. Perhaps ordering hardware and preparing the operating system build itself takes two months. After that, IT has to go through its standard testing process and operations procedures to make sure all the operations and support needs are identified. So, the earliest application delivery date would be in six months.

The business owner escalates the urgency of the issue, but cannot get past the process boundaries of the enterprise. Ultimately, the business owner outsources the project to a vendor and delivers the application in three months. Even though the application is delivered, it doesn't have the desired enterprise support and operations quality. It doesn't have to go this way—the company IT department should be the preferred and one-stop shop for all the business' needs. Even though outsourcing gives you a great return on investment, in the long run you lose significantly on the innovation. Your own IT department has the ability to innovate for you, and its members should be empowered to do so instead of being forced to overcome artificial process boundaries.

I see such scenarios on a daily basis, and I don't see a clear solution to the problem unless the entire process and structure in which these organizations operate is revamped, or unless technology like cloud computing is embraced wholeheartedly.

How will cloud computing help? To understand, let's go back to the original business requirement: the business owner has an immediate need to deploy an application, and the time frame is within three months. Basically, what the business is looking for is IT agility, and if the application takes only one month to develop, then is it really worth wasting six months on coordination and acquisition of the hardware?

Cloud computing gives you an instant-on infrastructure for deploying your applications. The provisioning of the hardware, operating system, and the software is all automated and managed by the cloud service providers.

Industry Terminology

For standardizing the overall terminology around cloud computing, the industry has defined three main cloud service categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

IaaS is a utility service that provides hardware and virtualized operating systems running in massively scalable data centers of the cloud service provider. You can then rent this infrastructure to deploy your own software and manage the life cycle of not only your software applications but also the underlying operating systems. In IaaS, you are still responsible for upgrading, patching, and maintaining the operating systems and the software applications that run on the rented hardware. Therefore, the target audiences for IaaS are system administrators and operations engineers. In short, IaaS abstracts the hardware and virtualization infrastructure from you.

PaaS is a utility service that provides the hardware, operating systems, and the runtime environment for running your applications in massively scalable data centers of the cloud service provider. The PaaS manages the operating systems and hardware maintenance for you, but you have to manage your applications and data. Therefore, naturally, the target audience for PaaS is typically developers. Even though the final deployment and maintenance will be managed by the operations teams, the platform empowers developers to make certain deployment decisions through configurations. In short, PaaS abstracts the infrastructure and the operating system from you.

SaaS is a utility service that provides you an end-to-end software application as a service. You only have to manage your business data that resides and flows through the software service. The hardware, operating systems, and the software are managed by the SaaS provider for you. Therefore, the target audience for SaaS is typically business owners who can go to the SaaS web site, sign-up for the service, and start using it.

In its natural progression, a SaaS is built on a PaaS and a PaaS is built on an IaaS. Therefore, PaaS providers have capabilities of IaaS built into PaaS. Whether to offer it as a separate service is mostly a strategic decision. Figure 7-1 illustrates the typical management boundaries between IaaS, PaaS and SaaS.

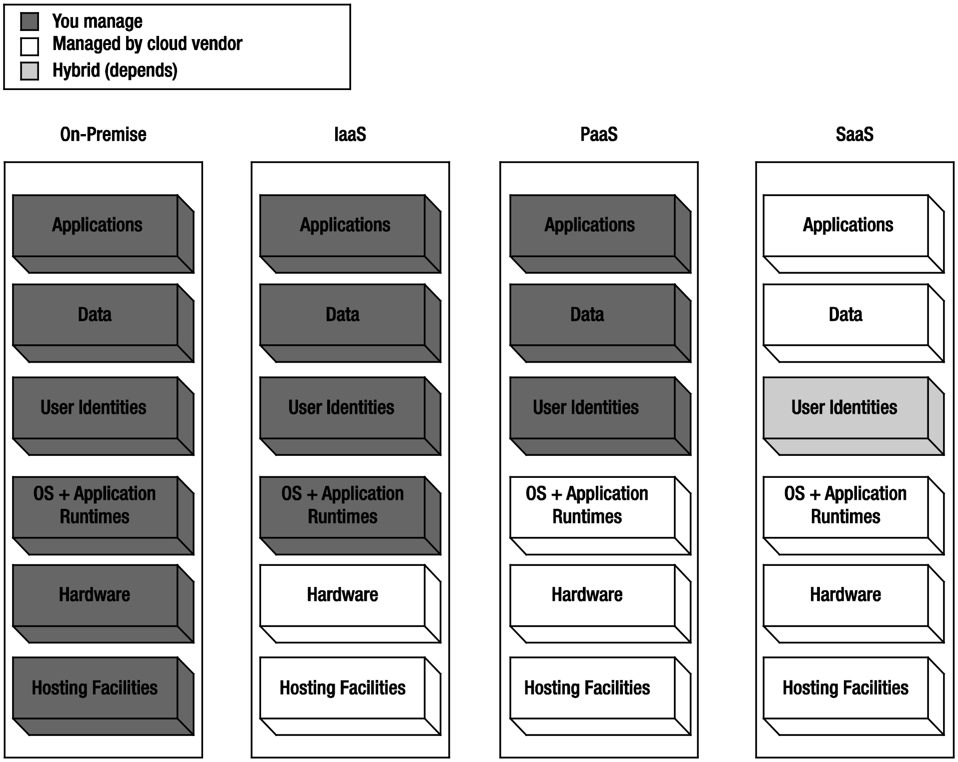


Figure 7-1. *IaaS, PaaS, and SaaS Management Boundaries*

The management of the user identities differs in different scenarios. Some enterprises will expose their on-premises identity provider as a federated service, while some businesses will keep SaaS and on-premises identities separate. Some additional terms that have been floating around in the past couple of years are Data as a Service (DaaS), IT as a Service, Security as a Service, and more. In the interest of simplicity, I have categorized all the services into IaaS, PaaS, and SaaS.

Types of Clouds

Along with the types of cloud services, the industry also frequently talks about the types of clouds that exist in the marketplace. A cloud is the underlying data center architecture that powers cloud services. Then what is the difference between a hosting provider and a cloud service provider? Great question....

From my experience and knowledge, I would define something as a cloud only if the data center architecture provides you with the following services:

- *Pay as you go service*: A cloud must provide you with a utility service model where you are charged only for the resources you use or by the number of users accessing the service. The price should go down or up dynamically based on your usage.

- *A self-service provisioning portal*: A cloud must provide you with a self-service portal for acquiring and releasing resources manually and programmatically.
- *Server hardware abstraction*: A cloud must relieve you from acquiring and/or maintaining any hardware resources required for an application to run.
- *Network hardware abstraction:* A cloud must relieve you from acquiring and/or maintaining any networking hardware resources required by your application.
- *Dynamic scalability*: A cloud must provide you with a manual and/or programmatic option for dynamically scaling your application up and down to match the demand.
- *High Availability Service Level Agreement (SLA)*: A cloud must clearly define an SLA with guaranteed availability of the platform.

The location of this cloud determines its type: private or public. In the interest of keeping the topic simple, I will define only these two types of clouds.

A public cloud is a data center that exists in the public domain and is accessible over the Internet. The public cloud is managed by the cloud service provider. Some public cloud platforms integrate with your company's intranet services through federation and virtual private networks or similar connectivity. The core application and the data still runs in the cloud service provider's data center.

A private cloud is cloud infrastructure running in your own data center. Because the definition of a cloud is wide open to interpretation, every company has carved out its own definition of a private cloud. I use the capabilities defined previously as bare minimum requirements for defining a public or a private cloud. If any of these services are not offered by a private cloud, then it's merely an optimized data center. And it is not necessarily a bad choice; an optimized data center may be a better choice than a cloud in some scenarios. The primary difference between private and public clouds is the amount of capital cost involved in provisioning infrastructure. Public clouds don't require provisioning.

■ **Note** I have used the terms *cloud services* and *cloud applications* interchangeably to generally represent cloud services. A cloud service may be thought of as a collection of cloud applications in some instances, but in our context, both mean the same thing.

Defining Our Terms

Before diving deep into cloud services, I would like to introduce you to the terminology I am using. Table 7-1 lists the terms and their definitions as I use them.

Table 7-1. Terminology

Term	Definition
Azure or Windows Azure	Microsoft's Windows Azure platform
Cloud application	An application deployed to a cloud services platform and typically part of a larger cloud service
Cloud platform	A PaaS offering by a cloud service provider for deploying cloud services (e.g., Windows Azure platform offered by Microsoft)
On-premise	Applications or services deployed and managed by an enterprise in its own data centers
Off-premise	Applications or services in the cloud
Solution	When used on its own, a collection of applications and/or cloud services designed for a specific business purpose (e.g., a payroll solution consisting of three cloud services and four on-premise applications)

Cloud Service Providers

In the past couple of years, several large software and Internet platform companies have started offering cloud services. It was a natural transition for companies like Amazon, Google, and Microsoft who already had a large Internet presence. VMware has been building these capabilities through acquisitions like Springsource and Zimbra. The offerings from all the cloud services are fragmented and it can sometimes be difficult to get a grasp of all the service offerings just from a single vendor. In Table7-2, I have listed a few providers with mature cloud services offerings. You can apply the same capabilities table to any cloud service provider present and future.

Table 7-2. Cloud Service Capabilities

Capability	IaaS	PaaS	SaaS
Public	Amazon EC2Windows Rackspace.com	Windows Azure platform Windows Azure AppFabric Force.com Google AppEngine	Office 365 Salesforce.com Google Apps
Private	VMWare vSphere Hyper-V	Windows Azure Appliance (Not yet available)	SharePoint as an IT service

From Table 7-2, you will be able to qualify the cloud service providers that fit your specific needs. Typically, you will not find any single cloud service provider that satisfies all your needs, which is true even with on-premises software.

Shifting to the Cloud Paradigm

As seen in the previous section, the choices provided by these offerings can put you in a dilemma, and most likely you will end up testing at least two cloud services before deciding on one. The move from a traditional on-premise model to an off-premise cloud model is a fundamental paradigm shift for businesses. Usually businesses are in their comfort zone when managing IT internally. With the cloud services model, even though the cost savings become evident, the challenge for businesses is to get out of their comfort zones and make the paradigm shift of moving to cloud services to stay competitive. The shift does not happen overnight; it takes several months of rigorous analysis, planning, and implementation. Depending on the costs, benefits, risks, and security requirements, a business can stay on-premise, embrace cloud services fully, or settle on a hybrid model yielding cost benefits while keeping core competencies on-site. Figure 7-2 illustrates the ownership of key enterprise assets in on-premise, cloud, and hybrid scenarios.

The recommended migration process is to move step by step, one application at a time. When the offshore software development model became popular in 2000, businesses faced a similar challenge in getting aboard the outsourcing wagon. Now, many businesses have significant offshore investments and clearly see the payoffs. It took time and learning for businesses to make the paradigm shift in offshore software development projects. For cloud services to succeed, businesses will be required to make a paradigm shift again.

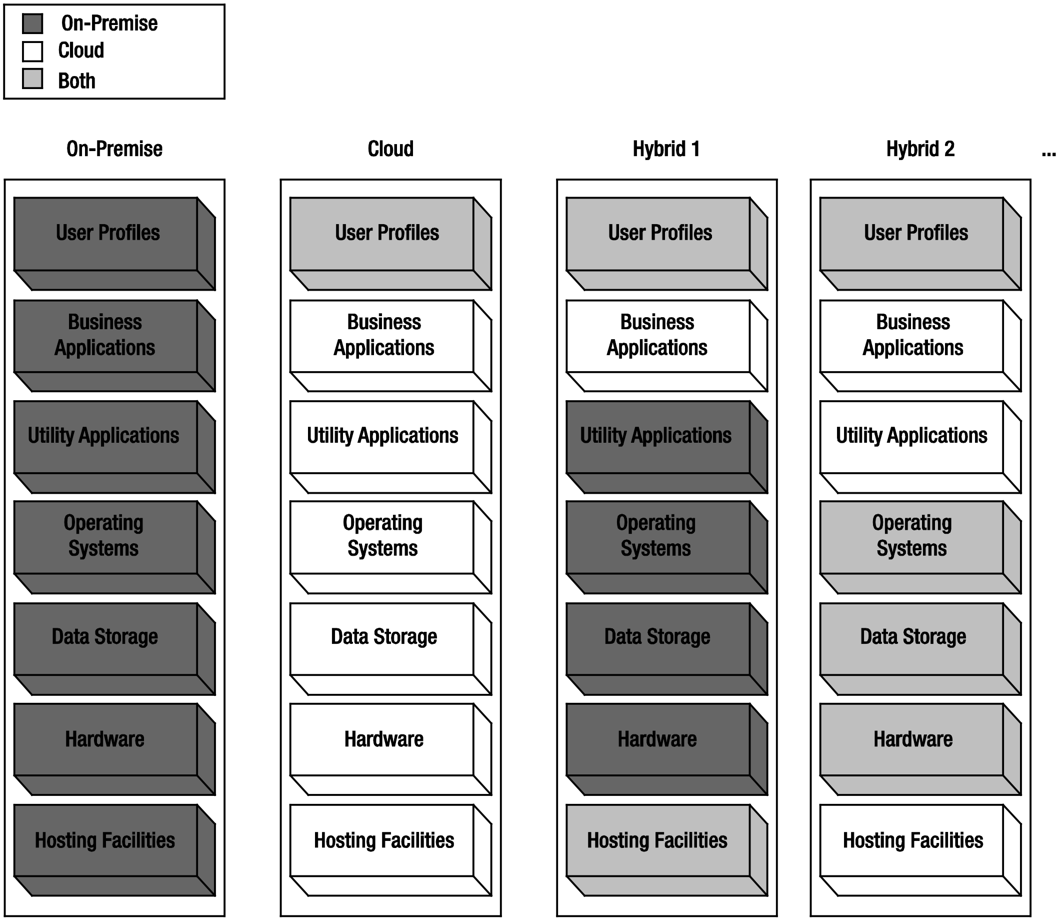


Figure 7-2. *On-premise, cloud, and hybrid scenarios*

In Figure 7-2, the on-premise and cloud scenarios are fairly easy to understand, because either all the assets are on-premise or in the cloud. The user profiles asset is usually required on both sides because of single sign-on requirements between on-premise and cloud services. In hybrid models, the businesses and the service provider must negotiate and decide which assets and services are better suited for locations on-premise, in cloud, or both. In the Hybrid 1 scenario, the user profiles and hosting facilities are present on both the sides; the business applications are in the cloud, whereas the utility applications, operating systems, data storage, and hardware are on-premise. In the Hybrid 2 scenario, the user profiles, operating systems, data storage, and hardware are present on both sides, whereas the business applications, utility applications, and hosting facilities are in the cloud. Most companies typically choose some hybrid model that best suits them.

Understanding the Cloud Services Ecosystem

The cloud services ecosystem consists of five major roles, as shown in Figure 7-3.

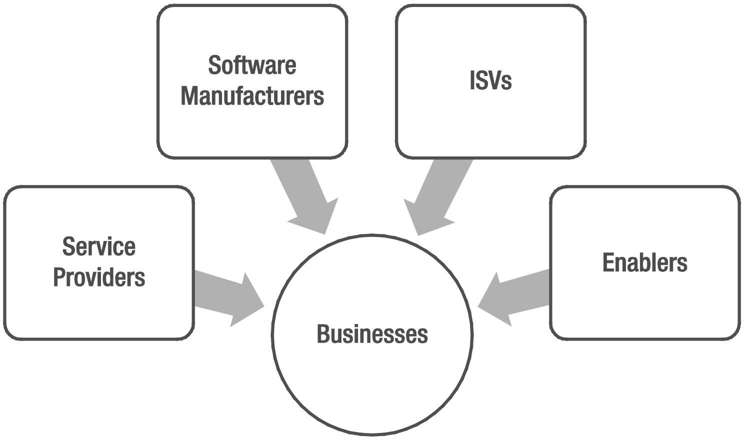


Figure 7-3. *The cloud services ecosystem*

Service Providers

The service providers are the companies that provide cloud services to the businesses and consumers. These companies run giant data centers hosting massively virtualized and redundant software and hardware systems. Service providers like Microsoft, with its Windows Azure platform, fall into this category. These providers not only have expertise in data center management, but also in scalable software management. The service providers may offer services directly to the businesses, consumers, or ISVs.

Software Vendors

Software designed to run on-premise is very different from software designed for cloud. Even though they both may provide the same business functionality to the end users, architecturally they may differ. The cloud services must account for multi-tenancy, scalability, reliability and performance at a much broader scale than on-premise architecture. Cloud services run in data centers offered by cloud service providers. In some cases, there is a significant overlap between the service providers and the software vendors. For example, the Microsoft Windows Azure platform and Microsoft's Office 365 are cloud software running in their own data centers. The software vendors have found it economically feasible to package hardware and software together in the data centers to optimize software delivery via cloud.

Independent Software Vendors

Independent software vendors (ISVs) are going to play a key role in the success of cloud services because of their expertise in vertical business applications. ISVs typically build vertical applications on an already existing platform. ISVs identify the business demand for a particular solution in vertical markets and thrive by offering the solution on existing platforms. The cloud offers a great platform for the ISVs to build vertical solutions. For example, an ISV could build a medical billing solution in the cloud and offer the service to multiple doctors and hospitals. The infrastructure required for building multitenant scalable software is already provided by the service providers, so the ISVs have to focus only on building the business solution and can enable them to penetrate new markets with lightning speed.

Enablers

Enablers (which are also called *implementers or system integrators*) are vendors offering services to build end-to-end solutions by integrating software from multiple vendors. Many enterprises purchase software licenses from vendors but never deploy the software because of lack of strategic initiative or availability of product expertise. Enablers fill in the gap by offering consulting services for the purchased software. Organizations like Microsoft Consulting Services offer customer-specific services regardless of the underlying platform. Enablers play a key role by integrating on-premise and cloud services or building end-to-end cloud services customized for a business. Cloud platform offers enablers an opportunity to expand their service offerings beyond on-premise solutions.

Businesses

Finally, businesses drive the demand for software products and services. If businesses see value or cost savings in a particular solution, they do not hesitate to implement it. To stay competitive in today’s market, businesses have to keep their IT and applications portfolios up to date and take advantage of economies of scale wherever possible. Cloud service offerings are architected to achieve economies of scale by supporting multiple businesses on a scalable and automated platform. For cloud service offerings to be successful, service providers, software vendors, ISVs, and enablers must work together in creating cloud applications and services not only providing cost savings but also a competitive edge to businesses and in-turn to consumers through these businesses.

Microsoft’s Cloud Strategy

For building a successful cloud services business, a company needs to first invest in building globally distributed data centers that are highly automated, efficient, and well connected. Building such data centers requires significant investment and support from software and operating system business partners to help monetize them. Therefore, typically, you will only see very large companies like Microsoft offer such services at a global scale.

Microsoft is the largest software manufacturer in the world, and its Global Foundation Services (GFS) group has done a phenomenal job in building a global network of data centers that can be leveraged by software partners within the company for delivering software services. This network of

Microsoft data centers is termed as Microsoft's cloud. I have toured one of these data centers, and I think they are among the most advanced in the world. What follows is a list of the 10 largest data centers in the world. Four in the top ten belong to Microsoft (I have highlighted them in bold).[1]

1. 350 East Cermak / Lakeside Technology Center (Digital Realty)
2. Metro Technology Center, Atlanta (Quality Technology)
3. The NAP of the Americas, Miami (Terremark)
4. NGD Europe, Newport Wales (Next Generation Data)
5. **Container Data Center, Chicago (Microsoft)**
6. **Microsoft Dublin (Microsoft)**
7. Phoenix ONE, Phoenix (i/o Datacenters)
8. CH1, Elk Grove Village, Ill. (DuPont Fabros)
9. **9A and 9B: Microsoft Data Centers in Quincy, Washington and San Antonio**
10. The SuperNAP, Las Vegas (Switch Communications)

As the adoption of cloud services increase in the IT industry, this list will change over time.

■ **Note** I highly recommend you visit `www.globalfoundationservices.com` for more information on Microsoft's data centers and GFS.

Microsoft's cloud strategy consists of the following four main initiatives:

1. Build a network of highly available data centers around the world as a software platform of the future.
2. Leverage these data centers for delivering its PaaS offerings.
3. Leverage these data centers for delivering its SaaS offerings.
4. Leverage the partner network for delivering IaaS offerings.

PaaS and SaaS offerings will be the primary mode of delivering most of its software assets in the future. In the past two years, Microsoft has successfully positioned Windows Azure platform as a leading public cloud platform. Microsoft went an extra mile innovating PaaS beyond traditional IaaS. Today, there are several Fortune 500 companies and small and midsize businesses actively using the Windows Azure platform. Ultimately Microsoft will need to provide an IaaS offering just to ease the on-boarding

[1] www.datacenterknowledge.com/special-report-the-worlds-largest-data-centers/

process for enterprises. The current on-boarding is not as attractive for enterprise customers because of investment required in migrating legacy applications to Windows Azure platform.

The Windows Azure Platform

In 2008 during the Professional Developer's Conference (PDC), Microsoft announced its official entry into the PaaS arena with the Windows Azure platform. Even though the SaaS offering called Business Productivity Online Suite (BPOS) or Office 365 has been around for a few years, the Windows Azure platform is an attempt to create a complete PaaS offering. See Figure 7-4.

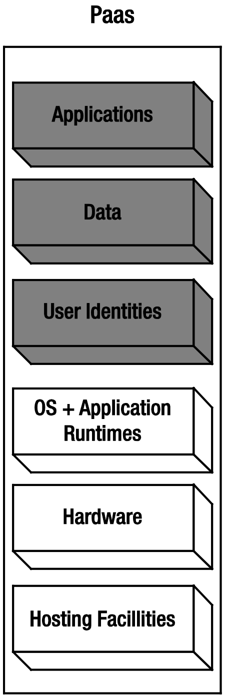
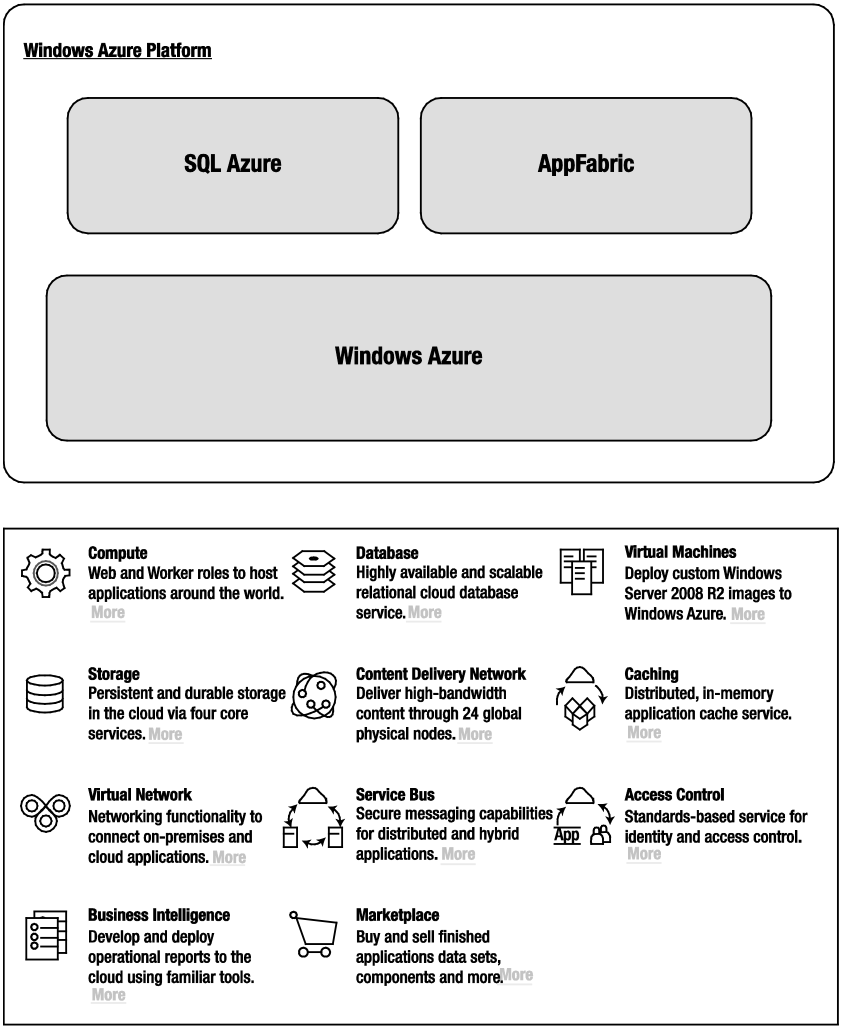


Figure 7-4. *Platform as a service*

Windows Azure platform is a key component of Microsoft's cloud strategy. The Windows Azure platform is a paradigm shift where unlimited resources are available at your fingertips for developing and deploying any .NET application in a matter of minutes. It disrupts your current ways of process-oriented sequential thinking. Microsoft has designed Windows Azure as an operating system for the data center. You don't have to wait for provisioning of any server or networking hardware, and then the operating system for deploying your application. Windows Azure platform drastically reduces the time from idea to production by completely eliminating the hardware and operating systems provisioning steps.

Windows Azure platform is a collection of building blocks for cloud services. Microsoft has been in the cloud business for quite some time with its consumer cloud services like MSN, Xbox Live, and Hotmail. Microsoft has also rebranded its business productivity and collaboration suite as Office 365 and includes services like SharePoint Online, Exchange Online, and Conferencing Services (Microsoft Office Lync). Windows Azure platform consists of three core components: Windows Azure, SQL Azure, and Windows Azure AppFabric, as shown in Figure 7-5.



Compute
Web and Worker roles to host applications around the world. More

Database
Highly available and scalable relational cloud database service. More

Virtual Machines
Deploy custom Windows Server 2008 R2 images to Windows Azure. More

Storage
Persistent and durable storage in the cloud via four core services. More

Content Delivery Network
Deliver high-bandwidth content through 24 global physical nodes. More

Caching
Distributed, in-memory application cache service. More

Virtual Network
Networking functionality to connect on-premises and cloud applications. More

Service Bus
Secure messaging capabilities for distributed and hybrid applications. More

Access Control
Standards-based service for identity and access control. More

Business Intelligence
Develop and deploy operational reports to the cloud using familiar tools. More

Marketplace
Buy and sell finished applications data sets, components and more.More

Figure 7-5. Microsoft Windows Azure platform (Source: `www.microsoft.com/windowsazure/features/`*)*

The core components are then split into several sub-components. But, the overall idea is to provide an a la carte menu so you can use any individual component in your solution. The pricing is typically based on each sub-component usage. The feature set is updated every six months, and there will be more features released soon after the time of writing.

■ **Note** Figure 7-5 only addresses the core components of the Windows Azure platform, as the platform matures; Microsoft is slowly promoting some sub-categories into its own offerings like the Virtual Network category that includes Windows Azure Connect and Traffic Manager. The Marketplace feature currently consists of the Windows Azure Marketplace DataMarket where you can publish and monetize your data over standard interfaces. Covering DataMarket is outside the scope of our coverage.

Windows Azure is the operating system for the data center that provides Compute, Storage, and Management services. SQL Azure is a relational database engine in the Windows Azure platform. Windows Azure AppFabric is the middleware component that consists of services like service bus, access control, and caching services. Developers can either build services that span across all these components or pick and choose the components as needed by the service architecture. The overall concept of Windows Azure platform is to offer developers the flexibility to plug in to the cloud environment as per the architectural requirements of the service.

If you acquire the full version of this book, *Window Azure Platform*, 2nd Edition, I cover all three main components and their sub-components.

Software development today typically consists one or more of the following types of applications:

- *Rich client and Internet applications*: Examples are Windows Client, Windows Presentation Foundation, HTML5, and Silverlight.
- *Web services and web applications*: Examples are ASP.NET, ASP.NET Web Services, and Windows Communications Foundation.
- *Server applications*: Examples are Windows Services, WCF, middleware, message queuing, and database development.
- *Mobile application*: Examples are .NET Compact Framework and mobile device applications.

The Windows Azure platform provides you with development tools and the deployment platform for developing and deploying all of these types of applications.

■ **Note** It took me more time to write this chapter than it did to develop and deploy a high-scale compute application on Windows Azure platform. Traditionally, the application would have taken 6–12 months to develop and deploy in production.

Understanding Windows Azure Compute Architecture

Fundamentally, the Windows Azure platform Compute architecture is based on a software fabric controller running in the data center and defining clear abstraction between server hardware and operating systems. The fabric controller automates the deployment of virtualized operating systems images on server hardware. In its simplest form, a typical cloud data center consists of a bank of server hardware and massive storage for storing fully functional operating system images. The fabric controller manages the life cycle of the deployment by allocating and decommissioning hardware and operating system images as needed. As a user, when you deploy your service to the cloud, the fabric controller provisions the hardware servers, deploys operating system image on those servers, provisions appropriate networking software like routers and load-balancers, and deploys your service to those servers. Once the service is deployed on servers, it is ready to be consumed. The numbers of service instances are configured by the service owner and would typically depend on the demand and high availability requirements of the service. Over the life cycle of the instance, the fabric controller is responsible for automating the maintenance, security, operations and high availability of the instances. Figure 7-6 illustrates the Windows Azure platform Compute architecture.

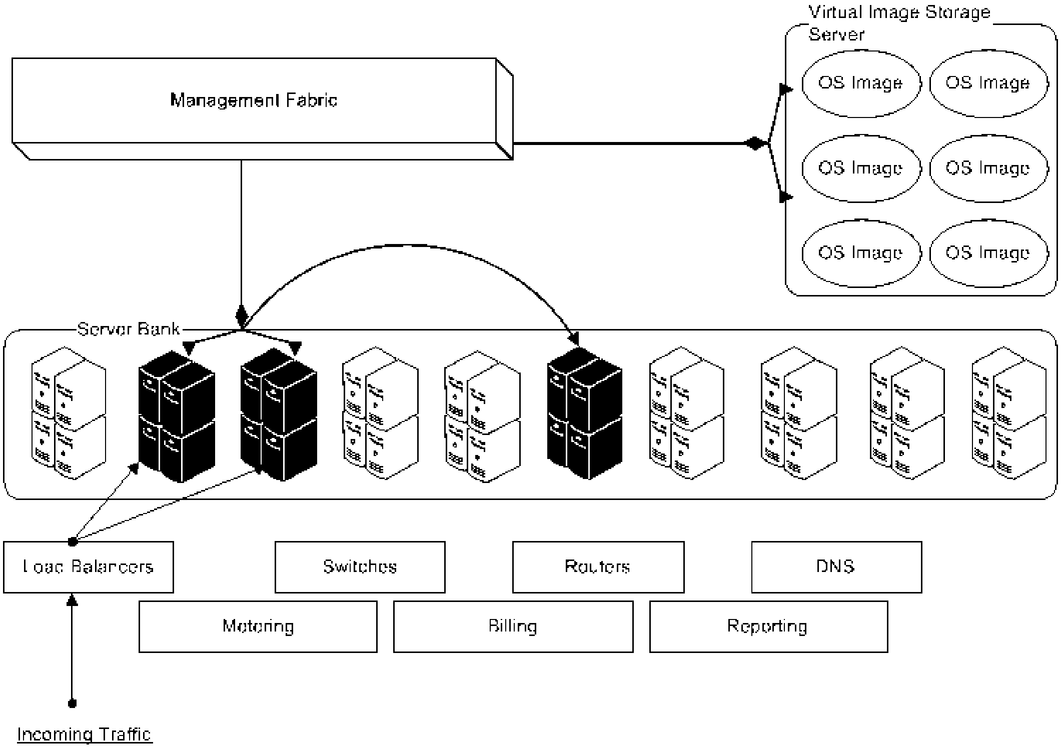


Figure 7-6. Windows Azure platform Compute architecture

The architecture also consists of some fixed assets like switches, routers, and DNS servers that manage the workload distribution across multiple service instances. This architecture is componentized and deployed into several geographically dispersed data centers for providing geo-located services. The metering, billing and reporting components complement the infrastructure with the ability to measure and report the usage of the service per customer.

■ **Note** Windows Azure platform is hosted in six data centers around the world. North Central US (1), South Central US (1), Europe (2), Asia (2). From the administration portal, you have the ability to choose the data center location for your application. Microsoft gives regular tours of these data centers to enterprises. You will have to poke your management to get a tour of one of these world-class data centers. You can find more information of these data centers at `www.globalfoundationservices.com`.

Windows Azure

Windows Azure is the core operating system of the platform that provides all the necessary features for hosting your services in the cloud. It provides a runtime environment that includes the IIS web server, background services, storage, queues, management services, and load-balancers. Windows Azure also provides developers with a local development fabric for building and testing services before they are deployed to Windows Azure in the cloud. Windows Azure also integrates seamlessly with the Visual Studio 2010 development environment by providing you with service publishing and IntelliTrace features. Figure 7-7 illustrates the three core services of Windows Azure.

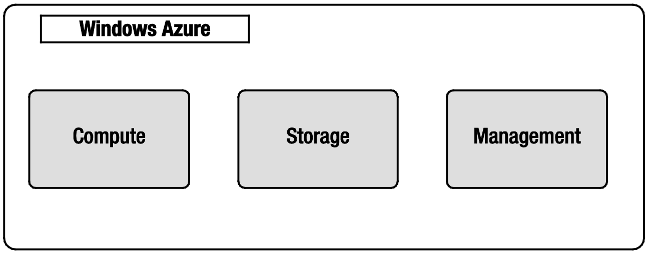


Figure 7-7. Windows Azure core services

Following is a brief description of the three core services of Windows Azure:

Compute – The Compute service offers scalable hosting of services on 64-bit Windows Server 2008 R2 at the time of this writing. The platform is virtualized and designed to scale dynamically based on demand. The platform runs Internet Information Server (IIS) version 7 enabled for ASP.NET Web applications. From version 1.3 of the SDK, you have access to Full IIS and administration features. You can also script start-up tasks that require administration privileges like writing to a registry or installing a COM dll library or installing third-party components like Java virtual machines. Developers can write managed and unmanaged services for hosting in Windows Azure Compute without worrying about the underlying operating systems infrastructure.

Storage – There are three types of storage supported in Windows Azure: tables, blobs, and queues. These storage types support direct access through REST APIs. Windows Azure tables are not a relational database like SQL Server tables. Instead, they provide structured data storage capabilities. They have an independent data model popularly known as the *entity* model. Tables are designed for storing terabytes of small-sized, highly available data objects. For example, user profiles in a high-volume ecommerce site would be a good candidate for tables. Windows Azure blobs are designed to store large sets of binary data like videos, images, and music in the cloud. Windows Azure queues are the asynchronous communication channels for connecting between services and applications not only in Windows Azure but also from on-premises applications. Queues are also the recommended method of communication between multiple Windows Azure role instances. The queue infrastructure is designed to support unlimited number of messages, but the maximum size of each message cannot exceed 8KB. Any account with access to storage can access tables, blobs, and queues. The total storage capacity of one storage account is 100TB and you can have multiple storage accounts. Windows Azure Drives provides NTFS drive volumes for Windows Azure applications in the cloud. So, you can create a drive, upload it to the blob storage and then attach it as an external drive to the Windows Azure instances. Drives provide you with durable storage access natively within your role but at the same time you lose the broad distributed nature of storing blobs that are scaled-out across multiple storage servers. A drive is still a single blob from Windows Azure Blob storage perspective.

Management – The Management service supports automated infrastructure and service management capabilities to Windows Azure cloud services. These capabilities include automatic commissioning of virtual machines and deploying services in them, as well as configuring switches, access routers, and load balancers for maintaining the user defined state of the service. The Management service consists of a fabric controller responsible for maintaining the health of the service. The fabric controller supports dynamic upgrade of services without incurring any downtime or degradation. The Windows Azure Management service also supports custom logging and tracing and service usage monitoring. You can interact with the Management service using a secure REST API. Most of the functionality available on the Windows Azure platform portal is available through the Service Management API for programmatically executing tasks. The API is widely used for automating the provisioning and dynamic scaling tasks. For example, the publish feature in the Visual Studio and the Windows Azure PowerShell cmdlets use the Service Management API behind the scenes to deploy cloud services to Windows Azure.

Figure 7-8 illustrates the Windows Azure architecture from an available service perspective.

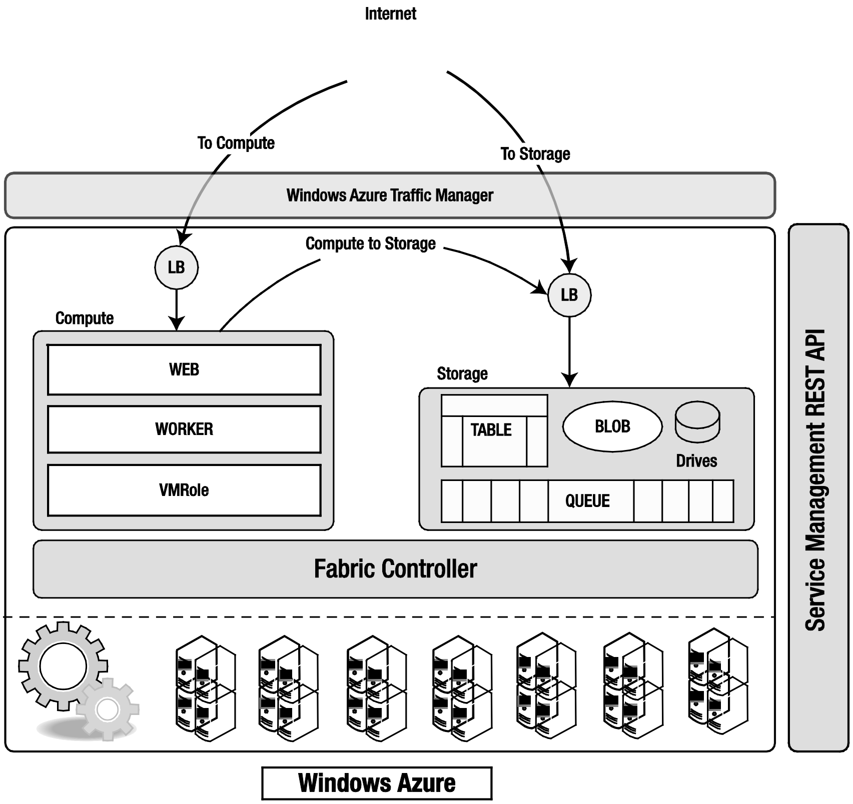


Figure 7-8. *Windows Azure*

When a request from the Internet comes in for your application, it passes through the load balancer and then to the specific Web/Worker role instance running your application. If a request for a storage service comes in, it passes through the load balancer to the appropriate storage service component.

Windows Azure Traffic Manager is a cross-region (data center) high-availability service that allows you to configure rules-based traffic diversions to your applications running in multiple Windows Azure data centers. Traffic Manager should be a key component in your disaster/recovery and business continuity strategy. Even though Traffic Manager manages traffic diversions, it does not replicate data across multiple data centers; you have to replicate data across multiple data centers for maintaining data consistency. At the time of writing, Traffic Manager was not available in production release. In the current version, there were three traffic diversion rules available: Fault-tolerance, Performance-based, and Round-Robin. In the CTP version, you could try it out for free.

Compute

The Windows Azure Compute service is based on a role-based design. To implement a service in Windows Azure, you have to implement one or more roles supported by the service. The current version of Windows Azure supports three roles: Web role, Worker role, and VM role. A role defines specific behavior for virtual machine instances running in the cloud. A web role is used for deploying web sites, a worker role is used for deploying background services or middle tier applications, and a VM role is typically used for running applications that do not fit a web or a worker role. Applications with intrusive installation process are well suited for VM role. Architecture of the system should dictate the types of roles you will need in your application.

The abstraction between the roles and the hardware is managed by the Fabric Controller. Fabric Controller manages end-to-end automation of the role instances, from hardware provisioning to maintaining service availability. Fabric Controller reads the configuration information you provide for your services and adjusts the deployment profile accordingly, as shown in Figure 7-9.

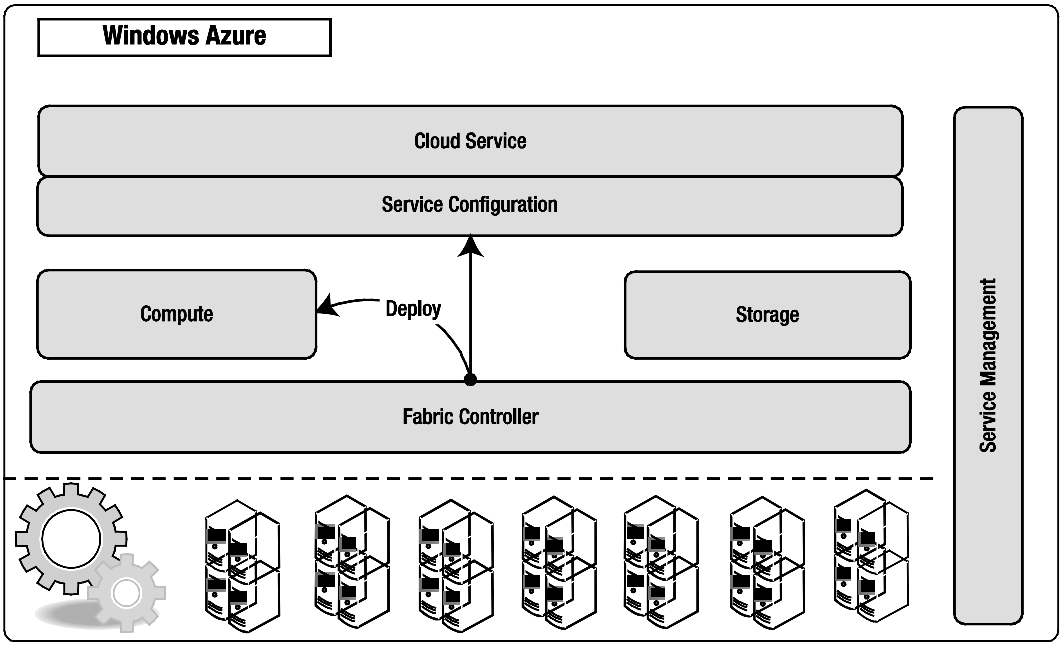


Figure 7-9. Fabric Controller deploys application

In the service configuration, you have to specify how many instances of a particular role you want to start with. The provisioning portal and the Service Management API can give you the status of the deployment. Once the cloud service is deployed, it is managed entirely by Windows Azure. You only manage your application and data.

Web Role

A Web role gives you the ability to deploy a web site or web service that can run in an IIS 7 environment. Most commonly, it will be an ASP.NET web application or external facing Windows Communications Foundation (WCF) service endpoints. Even though a Web role can host multiple sites, it is assigned only one external endpoint or entry point. But, you can configure different ports on the same entry point for http, https and custom TCP connections.

■ **Note** The UDP protocol is not supported at this time in Windows Azure services.

The Web role also supports FastCGI extension module to IIS 7.0. This allows developers to develop web applications in interpreted languages like PHP and native languages like C++. Windows Azure supports Full Trust execution that enables you to run FastCGI web applications in Windows Azure Web role. To run FastCGI applications, you have to set the `enableNativeCodeExecution` attribute of the Web role to `true` in the `ServiceDefinition.csdef` file. In support of FastCGI in the Web role, Windows Azure also introduces a new configuration file called `Web.roleconfig`. This file should exist in the root of the web project and should contain a reference to the FastCGI hosting application, like `php.exe`.

For more information on enabling FastCGI applications in Windows Azure, please visit the Windows Azure SDK site at `http://msdn.microsoft.com/en-us/library/dd573345.aspx`.

■ **Caution** Even though Windows Azure supports native code execution, the code still runs in the user context, not administrator, so some WIN32 APIs that require system administrator privileges will not be accessible by default but can be configured using the startup tasks and elevated privileges. I will cover start up tasks in detail in the next chapter.

Can I Run Existing Java Server Applications in Windows Azure?

There are a lot of enterprises that run Java; migrating these applications to Windows Azure is a big opportunity for Microsoft in winning and sustaining the underlying platform. Behind the scenes, Windows Azure runs Windows Server operating systems that can run Java virtual machines. Therefore, you can write custom scripts to install Java virtual machines on the Compute instances and then run your Java application on those instances. I have covered writing custom start-up tasks in Azure in a bit more detail in the next chapter, but the short answer is yes; you can run Java server applications on Windows Azure, but it's still not the first class citizen due to tools and endpoint limitations.

Worker Role

The Worker role gives you the ability to run a continuous background process in the cloud. It is analogous to Windows Services in the Windows platform. Technically, the only major difference between a Web role and a Worker role is the presence of IIS on the Web role. The Worker role can expose internal and external endpoints and also call external interfaces. A Worker role can also communicate with the queue, blob, and table storage services. A Worker role instance runs in a separate virtual machine from a Web role instance, even though both of them may be part of the same cloud service application. In some Windows Azure services, you may require communication between a Web role and a Worker role. Even though the Web and Worker roles expose endpoints for communication among roles, the recommended mode of reliable communication is Windows Azure queues. Web and Worker roles both can access Windows Azure queues for communicating runtime messages. The full book covers Windows Azure queues.

A Worker role class must inherit from the `Microsoft.WindowsAzure.ServiceRuntime.RoleEntryPoint` class. `RoleEntryPoint` is an abstract class that defines functions for initializing, starting and stopping the Worker role service. A Worker role can stop either when it is redeployed to another server, or you have executed the Stop action from the Windows Azure developer portal. Figure 7-10 illustrates the sequence diagram for the life cycle of a Worker role.

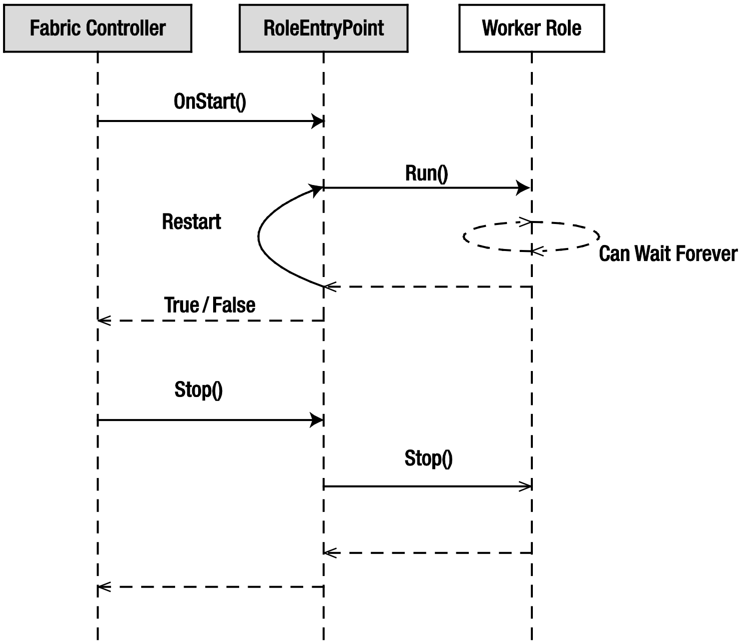


Figure 7-10. *Sequence diagram for a Worker role service*

In Figure 7-10, there are three objects: Fabric Controller, `RoleEntryPoint`, and a Worker role implementation of your code. Fabric Controller is a conceptual object; it represents the calls that the Windows Azure Fabric Controller makes to a Worker role application. The Fabric Controller calls the `Initialize()` method on the `RoleEntryPoint` object. `RoleEntryPoint` is an abstract class so it does not have its own instance; it is inherited by the Worker role instance to receive calls. The `OnStart()` method is a virtual method, so it does not need to be implemented in the Worker role class. Typically, you would write initialization code like starting diagnostics service or subscribing to role events in this method. The Worker role starts its application logic in the `Run()` method. The `Run()` method should have a continuous loop for continuous operation. If the `Run()` method returns, the role is restarted by the `OnStart()` method. If the role is able to start successfully, the `OnStart()` method returns `True` to the Fabric Controller; otherwise, it returns `False`. The Fabric Controller calls the `Stop()` method to shut down the role when the role is redeployed to another server or you have executed a Stop action from the Windows Azure developer portal.

VM Role

The VM role is specifically designed by Microsoft to reduce the barriers to entry into the Windows Azure platform. The VM role lets you customize a Windows Server 2008 R2 Enterprise virtual machine, based on a virtual hard drive (VHD), and then deploy it as a base image in Windows Azure. Typical scenarios for using a VM role are as follows:

- If you want to install software in your farm that does not support silent installation
- If the installation of a specific software component required manual intervention
- If you want to install a third-party application that needs significant modifications for deploying to the web or worker roles
- Any application that does not fit in Web role or a Worker role model
- Quick functional testing for specific software components that may later need to run on-premises

The VM role gives you more control over the software you can install on the virtual machine before uploading and running it in Windows Azure. Therefore, when creating a VM role, you have for first create a VHD on your local machine, install the appropriate software on it, and then upload the operating system image to Windows Azure. Once the operating system image is uploaded, you can then create a service definition for the service and then deploy multiple instances of the operating system image adhering to your service definition. In a Web role and a Worker role, the underlying operating system image is provided to you by Windows Azure, but in the case of a VM role, the service runs in the operating system image you created. Figure 7-11 illustrates the high-level process for creating a VM role image in Windows Azure. The complete book covers the entire process of creating and running VM roles in detail.

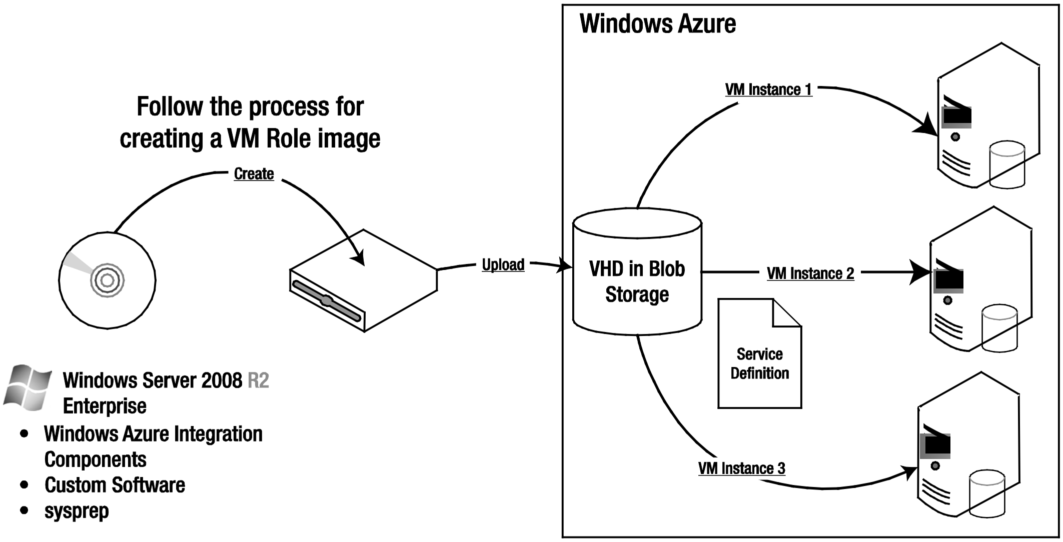


Figure 7-11. *Windows Azure VM Role*

■ **Tip** Don't use a VM role unless absolutely needed, because by acquiring more control over the underlying operating system image, you also inherit the risks and the burden associated with maintaining it. Windows Azure does not understand the health of your applications running on a VM role, and therefore it becomes your responsibility to track application health. For any Windows Azure application architecture, the Web role and the Worker role models must be preferred over the VM role.

Windows Azure Connect

Windows Azure Connect (aka Project Sydney) provides you with secure network connectivity between your on-premises machines and Windows Azure role instances. Windows Azure Connect is a new feature launched with Windows Azure SDK 1.3. With Windows Azure Connect, you can host your applications in Windows Azure and connect back to your data or applications that run on-premises. The motivation behind this feature was reducing the barriers to entry into Windows Azure. There is some data in enterprises that cannot be moved to the cloud due to various reasons like regulatory compliance, legal holds, company policies or simply IT politics. With Windows Azure Connect, the argument is reduced to tradeoff between latency and benefits of Windows Azure.

In the current version, Windows Azure Connect allows you to create point-to-point connectivity between your Windows Azure role instances and on-premises machines. It also gives you the ability to domain join the Windows Azure instances to your domain. Figure 7-12 illustrates the high-level architecture of the Windows Azure Connect capabilities.

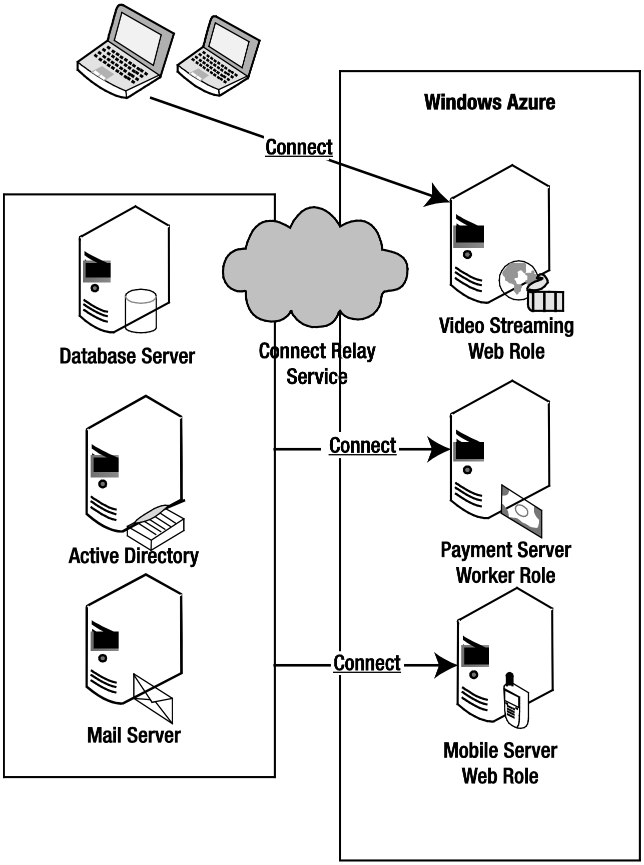


Figure 7-12. Windows Azure Connect

In Figure 7-12, the database server, the Active Directory server, and the mail server are grouped together for connecting with the Payment Server and Mobile Server Windows Azure role instances, whereas the development machines are grouped together for connecting with the Video Streaming role instance. The Windows Azure administration portal allows you to set up group-level as well as machine-level connectivity. The connectivity between the on-premises and Windows Azure instances is secured end to end via IPsec. Windows Azure Connect also uses a cloud-based relay service for firewall and NAT traversal of the network traffic between your on-premises machines and Windows Azure role instances.

■ **Note** In the current version (1.3), you cannot use Windows Azure Connect to connect between your Windows Azure role instances because they are assumed to connect with each other. For such communication, you can use Windows Azure queues, or input or internal endpoints.

■ **Tip** Don't use Windows Azure Connect unless absolutely needed because as you move data away from the application, you will see performance degradation due to latency. If you have an interactive web application, it is better to move the data closer to the application in Windows Azure Storage or SQL Azure. In some cases, you can separate the sensitive data and keep it on-premises and move rest of the data into the cloud and use Windows Azure Connect for calling on-premises applications.

Windows Azure Storage

Storage service offers the Compute nodes access to a scalable storage system. The Storage service has built-in high availability within a data center boundary. It maintains three copies of your data at any point in time behind the scenes. You can access the Storage service from anywhere through a REST API. The open architecture of the Storage service lets you design your applications to store data using REST APIs. Figure 7-13 illustrates the Windows Azure Storage service architecture.

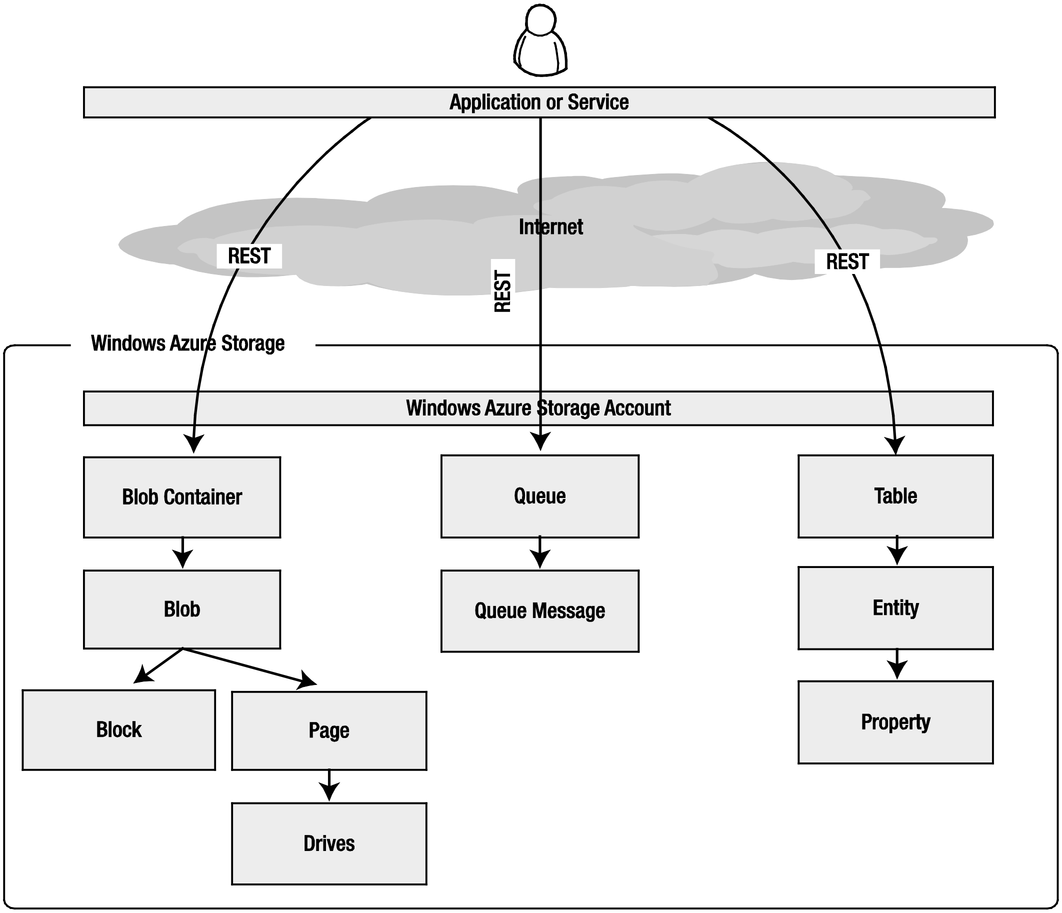


Figure 7-13. *Storage service architecture*

Windows Azure Storage supports four types of services: blobs, drives, queues, and tables. Blobs, queues, and tables have independent REST APIs. Drives are a special type of storage that is different from the other storage services. Drives are a type of page blob and are uploaded to the blob storage for mounting them to the Compute nodes. Drives don't have a direct REST API because they behave like other page blobs when uploaded to the blob storage. Drives do have a managed API in the SDK that you can use in your Compute nodes.

Windows Azure Storage types are scoped at the account level. This means that when you open a Storage account, you get access to all the Windows Azure Storage services: blobs, queues, and tables.

A blob account is a collection of containers. You can create any number of containers in an account. A container consists of number of blobs. A blob can be further composed of a series of blocks or pages.

A queue account is a collection of queues. An account can have any number of queues. A queue is composed of queue messages sent by the message sending applications.

Table 7-3 lists the commonalities and differences among the three storage types in Windows Azure.

Table 7-3. Windows Azure Storage

Feature	Blob	Queue	Table
URL Schema	http://[Storage Account] .blob.core.windows.net/	[Container Name] /[Blob Name]	http://[Storage Account] .queue.core.windows.net /[Queue Name] http://[Storage Account] .table.core.windows.net /[Table Name]?$ filter=[Query]
MAX Size	200GB(block blob)/1TB (page blob)	8K (string)	Designed for terabytes of data
Recommended Usage	Designed for large binary data types	Designed for cross-service message communication	Designed for storing smaller structured objects like the user state across sessions
API Reference	http://msdn.microsoft.com /en-us/library /dd135733.aspx	http://msdn.microsoft .com/en-us/library /dd179363.aspx	http://msdn.microsoft.com/ en-us/library/dd179423.aspx

Even though the Storage service makes it easy for Windows Azure Compute services to store data within the cloud, you can also access it directly from on-premises applications using the REST API. For example, you could write a music storage application that uploads all your MP3 files from you client machine to the blob storage, completely bypassing the Windows Azure Compute service. Compute and Storage services can be used independently of each other in Windows Azure. There are several customers using the Storage service purely for backing up their on-premises data to the cloud. You can create a scheduled automated utility that uploads back-up files to Windows Azure blob storage. This gives you clear separation between your production data and disaster recovery data.

■ **Note** The Windows Azure SDK also includes .NET-managed classes for calling the Storage service REST API. If you are programming in .NET, I recommend you to use the classes in Microsoft.WindowsAzure.StorageClient assembly.

Windows Azure, because it is a platform, does not provide you with any direct user interface for uploading files to the Storage service. You have to build your own application client for using the Storage service. There are several third-party tools like the Cloud Storage Studio from Cerebrata (`www.cerebrata.com/Products/CloudStorageStudio/Default.aspx`) you can use to upload and download files.

■ **Note** The Windows Azure Storage service is independent of the SQL Azure database service offered as part of the Windows Azure platform. Windows Azure Storage services are very specific to Windows Azure and unlike the Compute and SQL Azure services, there is no parity with any on-premises product offered by Microsoft.

Management

Unlike on-premise applications, the deployment of cloud services in PaaS involves only software provisioning from the developer's perspective. In a scalable environment for provisioning multiple services across thousands of instances, you need more programmatic control over the provisioning process. Manually uploading service packages and then starting and stopping services from the portal interface works well for smaller services, but are time-consuming and error-prone for large-scale services. The Windows Azure Service Management API allows you to programmatically perform most of the provisioning functions via a REST-based interface to your Windows Azure cloud account. The Service Management API is the hidden jewel of the platform. It makes Windows Azure a truly dynamically scalable platform allowing you to scale-up and scale-down your application on-demand. Using the Service Management API, you can automate provisioning, de-provisioning, scaling, and administration of your cloud services. Some of the common scenarios for leveraging the Service Management API are as follows:

- Automating the provisioning and de-provisioning of your cloud services through a well-defined release management process. Figure 7-14 illustrates the typical provisioning process of loading the applicationf package from blob store and deploying it through the Service Management API.

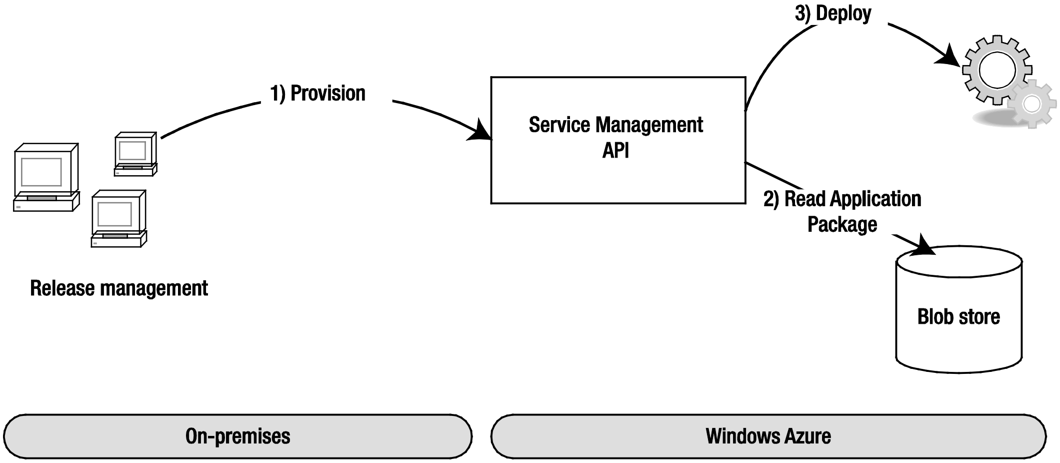


Figure 7-14. *Provisioning using the Service Management API*

- Dynamically scaling-up and down your applications based on the demand and application performance. Figure 7-15 illustrates a typical scaling architecture based on performance metrics of the roles instances of your application.

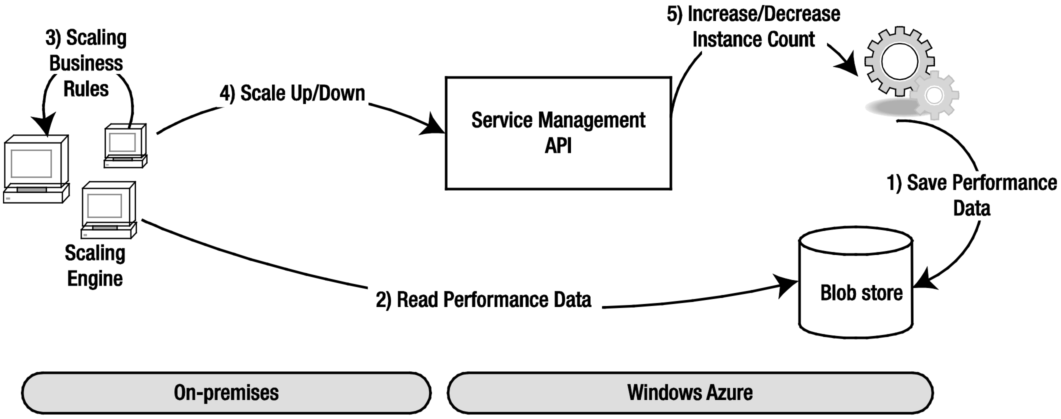


Figure 7-15. Scaling using the Service Management API

- The Service Management API also enables you to build an enterprise applications store in Windows Azure. Combining it with other Windows Azure platform services like Access Control and Windows Identity Foundation, you can federate on-premises identities with applications running in Windows Azure.

■ **Note** There are some third-party tools that offer dynamic scaling as a service. AzureWatch from Paraleap (`www.paraleap.com/AzureWatch`) is one such tool and even Microsoft Consulting Services has an Auto-scale toolkit that is offered in the context of a consulting engagement. You can also build your own using the Service Management API.

SQL Azure

SQL Azure is a relational database service in the Windows Azure platform. It provides core relational database management system (RDBMS) capabilities as a service, and it is built on the SQL Server product code base. In the current version, developers can access SQL Azure using tabular data stream (TDS), which is the standard mechanism for accessing on-premise SQL Server instances through the SQL client. The SQL client can be any TDS-compliant client, like ADO.NET, LINQ, ODBC, JDBC, or ADO.NET Entity Framework.

■ **Note** At the time of writing, the maximum database size allowed per database was 50GB. There are well-known partitioning patterns (e.g., sharding) for distributing data across multiple instances of 50GB databases. Built-in support for federation is on the roadmap for SQL Azure.

Figure 7-16 illustrates the core components of SQL Azure.

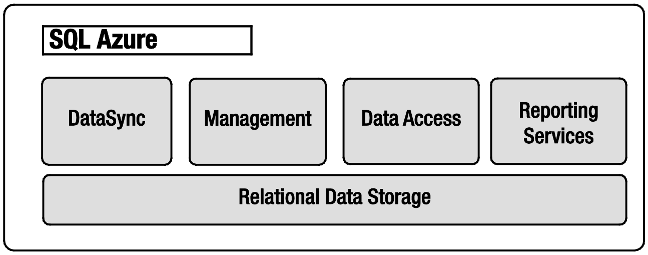


Figure 7-16. *SQL Azure core components*

The core services offered by SQL Azure are as follows:

Relational Data Storage – The relational data storage engine is the backbone of SQL Azure and is based on the core SQL Server code base. This component exposes the traditional SQL Server capabilities like the tables, indexes, views, stored procedures, and triggers. From a developer's perspective, SQL Azure is a relational subset of SQL Server. Therefore, if you have developed for SQL Server, you are already a SQL Azure developer. The majority of the differences are along the physical management of the data and some middleware features like ServiceBroker that are not offered in SQL Azure.

Data Sync – The Data Sync capabilities provide the synchronization and aggregation of data to and from SQL Azure to enterprise, workstations, partners and consumers devices using the Microsoft Sync Framework. The Data Sync component has two flavors: Data Sync service between SQL Azure instances and Data Sync services between on-premises database and SQL Azure. Both the services are based on Microsoft Sync Framework. With the combination of the two flavors, you can trickle down data all the way to field offices that are in remote locations.

■ **Note** The Microsoft Sync Framework is included in the SQL Server 2008 product (`http://msdn.microsoft.com/en-us/sync/default.aspx`).

Management – The Management component provides automatic provisioning, metering, billing, load-balancing, failover, and security capabilities to SQL Azure. Each database is replicated to one primary and two secondary servers. In case of a failover, the switching between the primary and the secondary server is automatic without interruptions. You can manage your SQL Azure databases from the Windows Azure portal as well as other existing toolsets like SQL Server Management Studio, OSQL, and BCP.

Data Access – The Data Access component defines different methods for accessing SQL Azure programmatically. Currently, SQL Azure will support Tabular Data Stream (TDS), which includes ADO.NET, Entity Framework, ODBC, JDBC, and LINQ clients. Developers can access SQL Azure either directly from on-premise applications or through cloud services deployed in Windows Azure. You can also locate a Windows Azure Compute cluster and a SQL Azure instance in the same data center for faster data access.

Reporting Services –The Windows Azure platform is new, and you should expect it to become richer in features every year. Microsoft is committed to reducing the parity gap between on-premises and Windows Azure platform services. SQL Azure is no exception; during PDC 2010, Microsoft announced the availability of SQL Reporting Services in Windows Azure platform. This brings reporting capabilities to your applications and an essential step towards business intelligence in the cloud.

Some of the common scenarios for leveraging the SQL Azure are as follows:

Database consolidation – If you have home-brewed databases that are sitting under desktops or on isolated islands, then having SQL Azure as one of the options in your database consolidation strategy will greatly help you in achieving your long-term goal of reducing the operations footprint. You can migrate to SQL Azure from Access, MySQL, SQL Server, DB2, Oracle, Sybase, and pretty much any other relational database to SQL Azure with proper tools like SQL Server Migration Assistant (SSMA) and SQL Azure Migration Wizard from Codeplex. If there is no direct migration path to SQL Azure for a particular database, then the most common path is from third-party database to SQL Server and then from SQL Server to SQL Azure. SQL Azure will provide you with ability to quickly provision databases and reduce the time to market for your solution. You don't have to wait for provisioning clustered hardware for running your databases.

Back-end for Windows Azure Compute – SQL Azure is the recommended relational database for applications running in Windows Azure Compute. If you co-locate your Windows Azure Compute instances and SQL Azure in the same data center, you can get great performance advantage and you will also not incur any data-transfer costs.

Geo-replication of Data – SQL Azure and Data Sync services for SQL Azure give you the ability to synchronize two-way synchronizations across multiple geographies and also trickle down data to remote hubs. This kind of data movement allows you to move data closer to the application for better user experience. Figure 7-17 illustrates a common geo-replication pattern in SQL Azure.

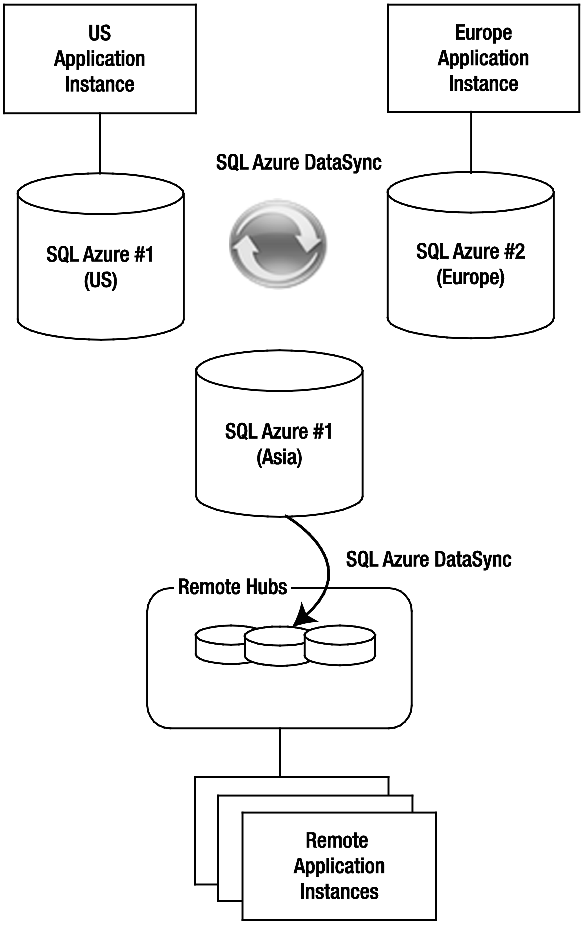


Figure 7-17. Geo-replication in SQL Azure

Quick Application Migrations – SQL Azure allows you to quickly migrate the relational database component of your on-premises application to Windows Azure. This is a big differentiator for Microsoft in comparison to its competitors. With minimal efforts, you can quickly move your existing relational data to Windows Azure and then focus on migrating the application. Once the data is moved to SQL Azure, you can just change the connection string of your application to point to the SQL Azure database.

■ **Tip** SQLAzure runs a labs program where you can try out upcoming features for free and provide feedback to the Microsoft product teams. You can find more information about the labs program here `www.sqlazurelabs.com`.

Windows Azure AppFabric

Windows Azure AppFabric is the cloud-based middleware platform hosted in Windows Azure. Windows Azure AppFabric is the glue for connecting critical pieces of a solution in the cloud and on-premises. You must have heard several times from industry speakers and even cloud enthusiasts that "everything is moving to the cloud." I firmly believe that everything is *not* moving to the cloud. The disruptive effect of cloud computing is similar to the disruption created by cell phones. Cell phones gave users quick provisioning and anywhere access to communications. But, some critical functions like building security still run over landlines and have no plans for switching to a cell phone network. Windows Azure AppFabric provides you with components for integrating Windows Azure applications with on-premises applications. The core components of the Windows Azure AppFabric are as follows:

- An Internet service bus, called Service Bus, for connecting applications running in Windows Azure (and other domains) and on-premises
- A claims-mapping service, called Access Control Service, for supporting claims-based authorization in your applications running in Windows Azure and on-premises
- A distributed caching service within the Windows Azure platform, called Windows Azure AppFabric Cache

■ **Note** Windows Azure AppFabric and Windows Server AppFabric are two different products. In the long run, Microsoft's goal is to bring services available in Windows Server AppFabric to Windows Azure AppFabric. Some of the features like ACS and Service Bus will always remain exclusive to Windows Azure AppFabric.

I think of Azure AppFabric as the integration middleware of the Windows Azure platform, because it provides connectivity, caching, identity claims federation, and messaging capabilities among distributed applications. You can leverage these capabilities not only for cloud services but also for on-premises applications. Microsoft's strategy is to provide Windows Azure AppFabric as a middleware and building blocks for building and deploying distributed applications on Windows Azure. Figure 7-18 illustrates the three core services of Windows Azure AppFabric.

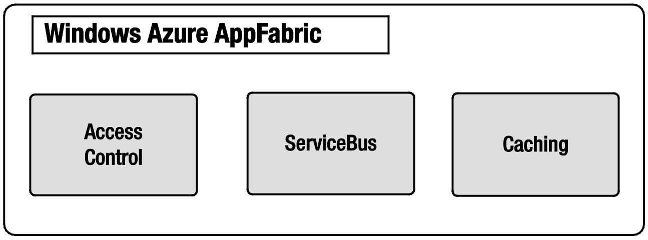


Figure 7-18. *Windows Azure AppFabric core services*

Access control – The access control service provides rules-driven, claims-based access control for distributed applications. The access control service is designed to abstract identity providers from your application. It maps input claims from different types of identity providers to a standard set of output claims known to your application. This architecture provides a model for dynamically adding and removing identity providers without changing a single line of code in your application. The most common uses of Access Control service are:

- Providing identity federation between multiple partner identity providers and partner portals
- Providing enterprise identity federation using ADFS 2.0 and enterprise applications running in Windows Azure

Service bus – The service bus is a generic Internet service bus based on the Windows Communications Foundations (WCF) programming model. It is analogous to the Enterprise Service Bus (ESB) popularly seen in large enterprises. Unlike the ESB, the Azure AppFabric Service Bus is designed for Internet scale and messaging with cross-enterprise and cross-cloud scenarios in mind. The service bus provides key messaging patterns like publish/subscribe, point-to-point, and durable buffers for message exchanges across distributed applications in the cloud as well as on-premise. You can expose an on-premises line-of-business application interface as a Service Bus endpoint and then consume that endpoint from a Windows Azure application or any other application. The most common uses of Service Bus are:

- Accessing line-of-business data from Windows Azure applications
- Providing connectivity to data residing in on-premises applications at the web service level (Windows Azure Connect provides connectivity at the network level)

Caching – The caching service was announced during PDC 2010. It is a late comer to the Azure AppFabric family, but one of the most valuable. The caching service provides distributed caching for applications running Windows Azure. Caching is an essential component for any internet-scale application. Before the caching service, developers either built custom caching components or modified third-party components like memcached (http://memcached.org/)to run on Windows Azure. The Azure AppFabric caching service makes caching a first-class citizen of the Windows Azure platform. It provides in-memory as well as external data center co-located caching service. The most common uses of the caching service in Windows Azure are:

- In-memory caching of SQL Azure database data
- Caching on-premises data to be consumed by Windows Azure applications
- Caching multi-tenant user interface data for providing high-performance interactive applications
- Caching user state in multi-player gaming applications
- Caching location-based usage information

- Caching session state of users

■ **Tip** Like SQL Azure, Windows Azure AppFabric also runs a labs program where you can try upcoming features for free. You can login and start playing with these features here `https://portal.appfabriclabs.com`.

Now that we have covered the Windows Azure technology fundamentals, let's see how the pricing for all these features is structured. In the cloud, every feature you use is metered based on usage like your electricity bill, therefore architects and developers needs to choose appropriate features and are naturally forced to not over-architect an application.

Windows Azure Platform Pricing

Each Windows Azure platform component is priced differently and within each component there are further pricing choices. Typically, architects are used to designing applications assuming capital investment for supporting maximum capacity. But, cloud platforms give you the flexibility for designing applications for minimum capacity and scale up dynamically based on demand. Such flexibility adds a new variable "operating cost" to your design. Every cloud resource you plan to utilize in your design has a cost associated with it. Cloud computing gives rise to a new architecture paradigm I call cost-driven architecture (CDA). In CDAs, an architect iteratively evaluates the operating cost of the architecture and modifies the architecture according to the cost boundaries of the application as shown in Figure 7-19.

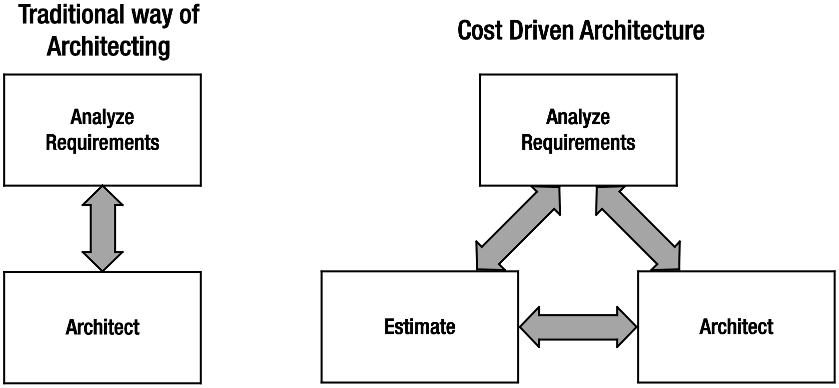
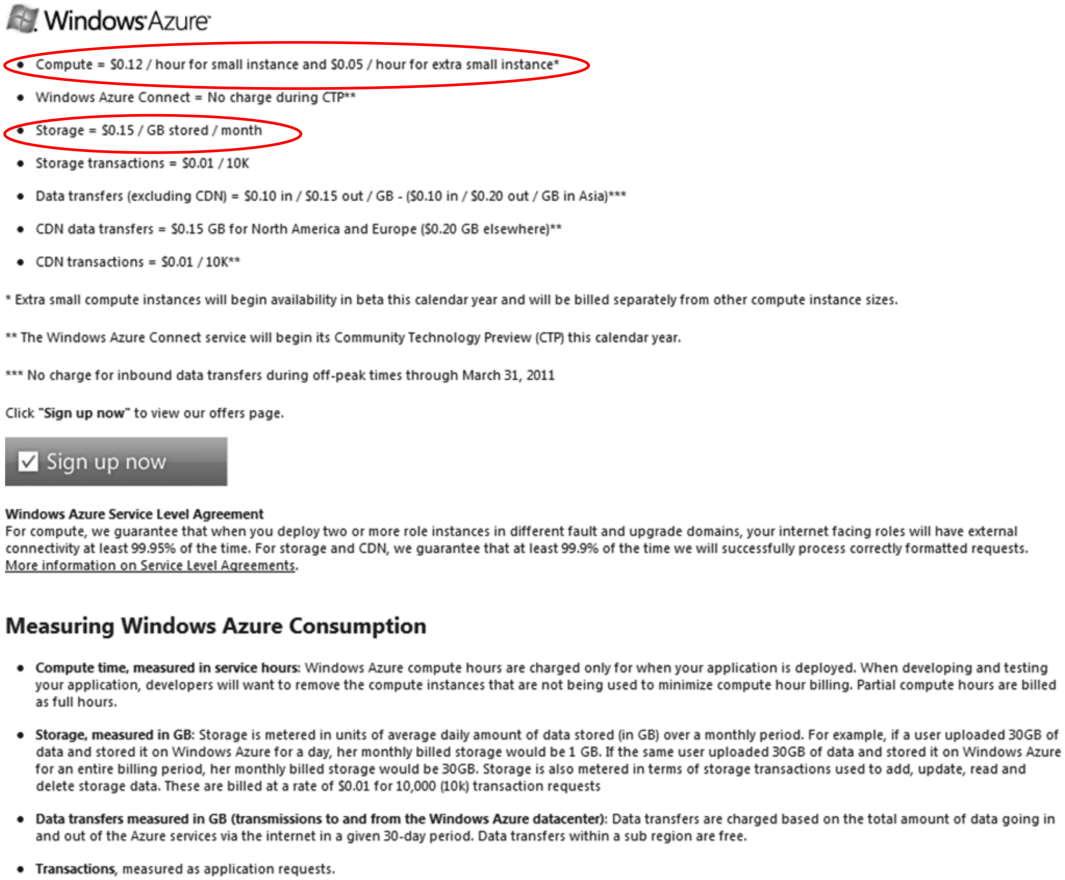


Figure 7-19. *Cost-driven architecture*

CDAs are tied directly to the pricing of the cloud services. The pricing for Windows Azure platform components are available publicly at:

`www.microsoft.com/windowsazure/pricing/default.aspx`

Figure 7-20 illustrates the pricing for Windows Azure Compute and Storage services.



- Compute = $0.12 / hour for small instance and $0.05 / hour for extra small instance*
- Windows Azure Connect = No charge during CTP**
- Storage = $0.15 / GB stored / month
- Storage transactions = $0.01 / 10K
- Data transfers (excluding CDN) = $0.10 in / $0.15 out / GB - ($0.10 in / $0.20 out / GB in Asia)***
- CDN data transfers = $0.15 GB for North America and Europe ($0.20 GB elsewhere)**
- CDN transactions = $0.01 / 10K**

* Extra small compute instances will begin availability in beta this calendar year and will be billed separately from other compute instance sizes.

** The Windows Azure Connect service will begin its Community Technology Preview (CTP) this calendar year.

*** No charge for inbound data transfers during off-peak times through March 31, 2011

Click "**Sign up now**" to view our offers page.

Windows Azure Service Level Agreement
For compute, we guarantee that when you deploy two or more role instances in different fault and upgrade domains, your internet facing roles will have external connectivity at least 99.95% of the time. For storage and CDN, we guarantee that at least 99.9% of the time we will successfully process correctly formatted requests. More information on Service Level Agreements.

Measuring Windows Azure Consumption

- **Compute time, measured in service hours:** Windows Azure compute hours are charged only for when your application is deployed. When developing and testing your application, developers will want to remove the compute instances that are not being used to minimize compute hour billing. Partial compute hours are billed as full hours.
- **Storage, measured in GB:** Storage is metered in units of average daily amount of data stored (in GB) over a monthly period. For example, if a user uploaded 30GB of data and stored it on Windows Azure for a day, her monthly billed storage would be 1 GB. If the same user uploaded 30GB of data and stored it on Windows Azure for an entire billing period, her monthly billed storage would be 30GB. Storage is also metered in terms of storage transactions used to add, update, read and delete storage data. These are billed at a rate of $0.01 for 10,000 (10k) transaction requests
- **Data transfers measured in GB (transmissions to and from the Windows Azure datacenter):** Data transfers are charged based on the total amount of data going in and out of the Azure services via the internet in a given 30-day period. Data transfers within a sub region are free.
- **Transactions**, measured as application requests.

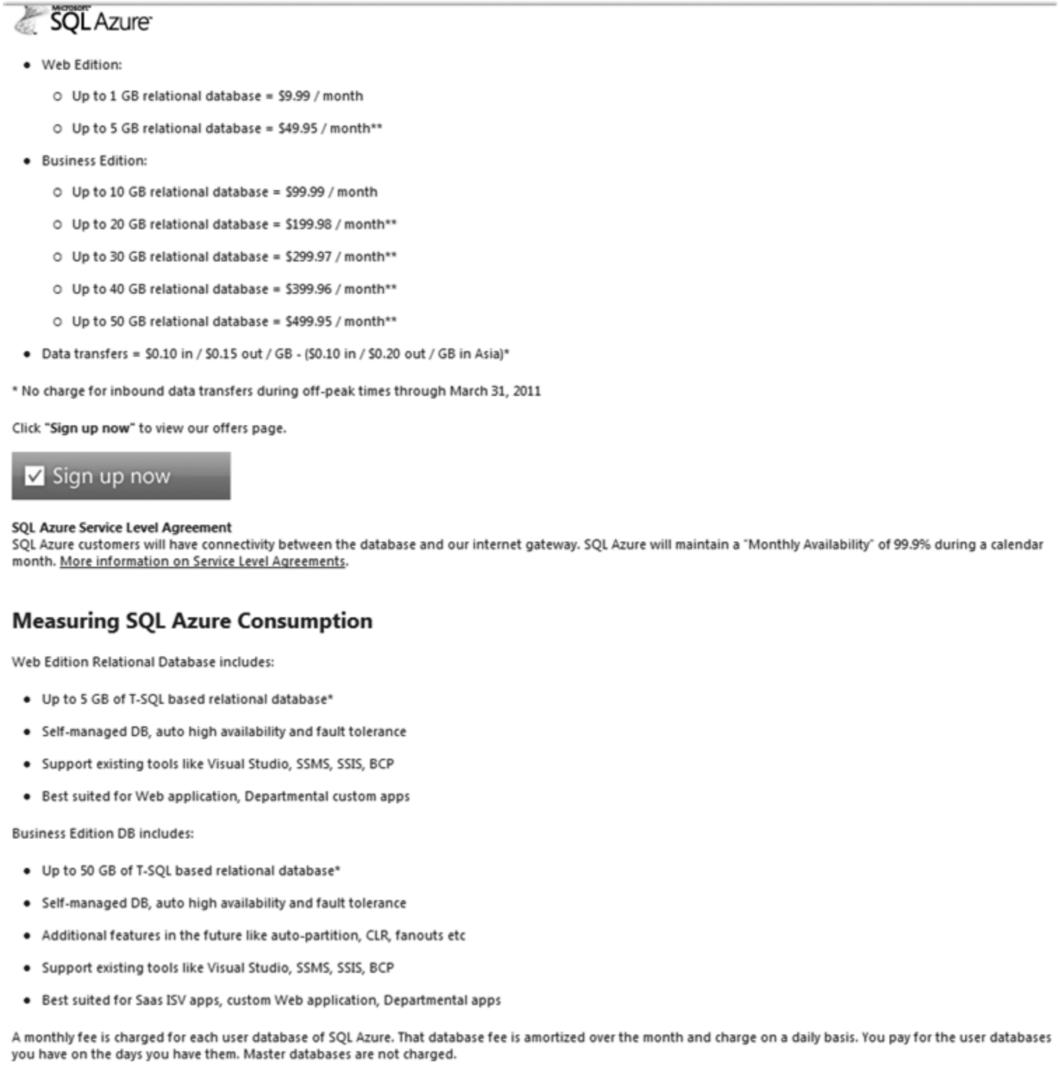
Figure 7-20. *Windows Azure Compute and Storage pricing*

A couple of things to observe in Figure 7-20 are as follows:

- The Compute service charges are per hour of consumption and the price is based on the size of the virtual machine instance. You will be charged not only for a running application but also for an application that is deployed but not running. You have to unload the application to avoid any charges.
- The Storage service charges are for data storage per GB as well as transactions (ingress and egress) in and out of the data center. So, if your Compute instances and storage are located in the same data center, there will be no transaction cost for data transfer between the two, but if the Compute instances access Storage service from another data center, you will incur transaction costs.

■ **Note** Microsoft no longer charges for input data bandwidth (ingress). There is network bandwidth charge for only egress (data-out); therefore it is easier to migrate your data into Windows Azure.

Figure 7-21 illustrates the pricing for SQL Azure.



- Web Edition:
  - Up to 1 GB relational database = $9.99 / month
  - Up to 5 GB relational database = $49.95 / month**
- Business Edition:
  - Up to 10 GB relational database = $99.99 / month
  - Up to 20 GB relational database = $199.98 / month**
  - Up to 30 GB relational database = $299.97 / month**
  - Up to 40 GB relational database = $399.96 / month**
  - Up to 50 GB relational database = $499.95 / month**
- Data transfers = $0.10 in / $0.15 out / GB - ($0.10 in / $0.20 out / GB in Asia)*

* No charge for inbound data transfers during off-peak times through March 31, 2011

Click **"Sign up now"** to view our offers page.

SQL Azure Service Level Agreement
SQL Azure customers will have connectivity between the database and our internet gateway. SQL Azure will maintain a "Monthly Availability" of 99.9% during a calendar month. More information on Service Level Agreements.

Measuring SQL Azure Consumption

Web Edition Relational Database includes:

- Up to 5 GB of T-SQL based relational database*
- Self-managed DB, auto high availability and fault tolerance
- Support existing tools like Visual Studio, SSMS, SSIS, BCP
- Best suited for Web application, Departmental custom apps

Business Edition DB includes:

- Up to 50 GB of T-SQL based relational database*
- Self-managed DB, auto high availability and fault tolerance
- Additional features in the future like auto-partition, CLR, fanouts etc
- Support existing tools like Visual Studio, SSMS, SSIS, BCP
- Best suited for Saas ISV apps, custom Web application, Departmental apps

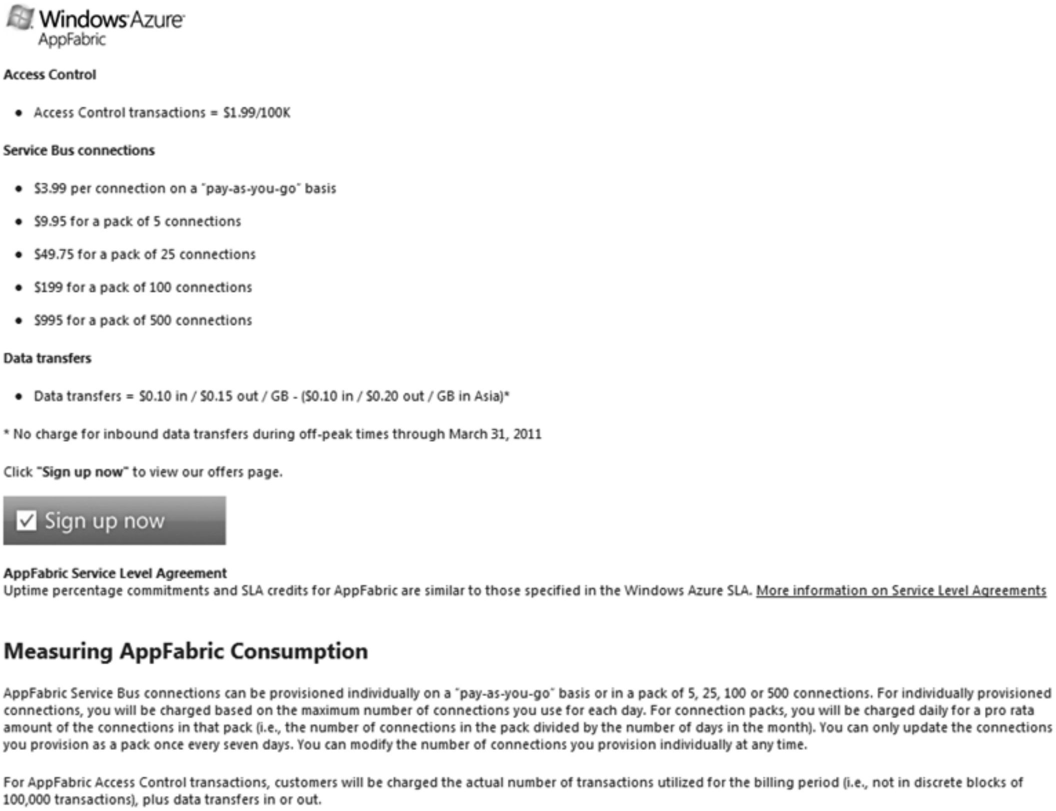
A monthly fee is charged for each user database of SQL Azure. That database fee is amortized over the month and charge on a daily basis. You pay for the user databases you have on the days you have them. Master databases are not charged.

Figure 7-21. SQL Azure pricing

A few things to observe in Figure 7-21 are:

- There are two types of database editions: Web and Business (10GB or more is the Business Edition).
- The maximum size per database is 50GB, but you can create multiple 50GB databases and partition data between those instances.
- "The database fee is amortized over the month and charged on a daily basis." If you dynamically create and delete databases frequently, you have to make sure you factor in this daily cost.
- Similar to Storage service, SQL Azure also charges for data transfer between data centers. There will be no charge for data transfer within the same data center.

Figure 7-22 illustrates the pricing for Windows Azure AppFabric.



Access Control

- Access Control transactions = $1.99/100K

Service Bus connections

- $3.99 per connection on a "pay-as-you-go" basis
- $9.95 for a pack of 5 connections
- $49.75 for a pack of 25 connections
- $199 for a pack of 100 connections
- $995 for a pack of 500 connections

Data transfers

- Data transfers = $0.10 in / $0.15 out / GB - ($0.10 in / $0.20 out / GB in Asia)*

* No charge for inbound data transfers during off-peak times through March 31, 2011

Click "**Sign up now**" to view our offers page.

☑ Sign up now

AppFabric Service Level Agreement
Uptime percentage commitments and SLA credits for AppFabric are similar to those specified in the Windows Azure SLA. More information on Service Level Agreements

Measuring AppFabric Consumption

AppFabric Service Bus connections can be provisioned individually on a "pay-as-you-go" basis or in a pack of 5, 25, 100 or 500 connections. For individually provisioned connections, you will be charged based on the maximum number of connections you use for each day. For connection packs, you will be charged daily for a pro rata amount of the connections in that pack (i.e., the number of connections in the pack divided by the number of days in the month). You can only update the connections you provision as a pack once every seven days. You can modify the number of connections you provision individually at any time.

For AppFabric Access Control transactions, customers will be charged the actual number of transactions utilized for the billing period (i.e., not in discrete blocks of 100,000 transactions), plus data transfers in or out.

Figure 7-22. *Windows Azure AppFabric pricing*

■ **Tip** For quick cost estimation, there is a nice pricing calculator on the Windows Azure web site at `www.microsoft.com/windowsazure/pricing-calculator/`. I encourage you to try it out by mapping any of your on-premise application to Windows Azure.

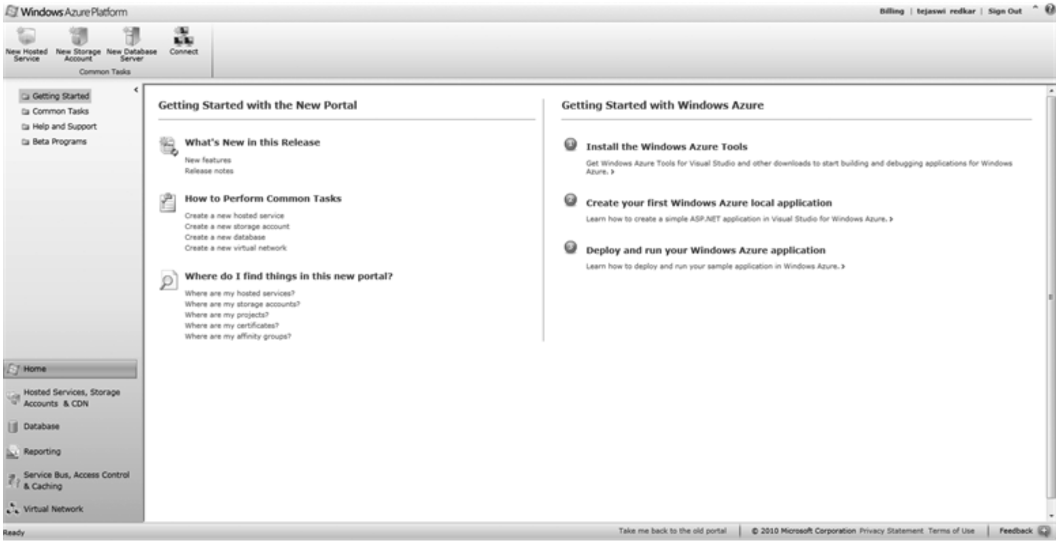
All of the Windows Azure platform components can be managed from the Windows Azure platform Management portal. Let's review that now.

Management Portal – Let's Provision

■ **Note** The portal experience will be much different from what it is at the time of writing, but the concepts will still remain the same.

Microsoft has designed the Windows Azure platform Management portal in Silverlight, providing better interactivity and a central place for managing all the Windows Azure platform components. You can access the portal via the following URL https://windows.azure.com. Before accessing the portal, you will need to create a subscription using your Live Id. Microsoft offers several introductory specials that can be found here `www.microsoft.com/windowsazure/offers/`. Sometimes there are one month free trials. If you have MSDN subscription, you also receive some free hours per month. After you create a subscription, you can log in to the Management portal using the same Live Id you used for creating the subscription.

Once you create a subscription, you get access to all the Windows Azure platform components. After you log in to the Management portal, you will be taken to the main portal page. Figure 7-23 illustrates the screenshot of the main portal page.



Getting Started with the New Portal

What's New in this Release

New features
Release notes

How to Perform Common Tasks

Create a new hosted service
Create a new storage account
Create a new database
Create a new virtual network

Where do I find things in this new portal?

Where are my hosted services?
Where are my storage accounts?
Where are my projects?
Where are my certificates?
Where are my affinity groups?

Getting Started with Windows Azure

Install the Windows Azure Tools

Get Windows Azure Tools for Visual Studio and other downloads to start building and debugging applications for Windows Azure. ›

Create your first Windows Azure local application

Learn how to create a simple ASP.NET application in Visual Studio for Windows Azure. ›

Deploy and run your Windows Azure application

Learn how to deploy and run your sample application in Windows Azure. ›

Figure 7-23. Windows Azure platform Management portal

The Management portal user interface is highly interactive and gives you the feeling of interacting with the desktop software. The left navigation lists all the services from the Windows Azure platform and the top navigation bar lists commands in the context of your navigation. For example, on the main page you can create new services: Hosted Service, Storage Service, Database Server, and Windows Azure Connect service. When you navigate to the Database tab, you will see the commands change in the context of the SQL Azure database server, as shown in Figure 7-24.

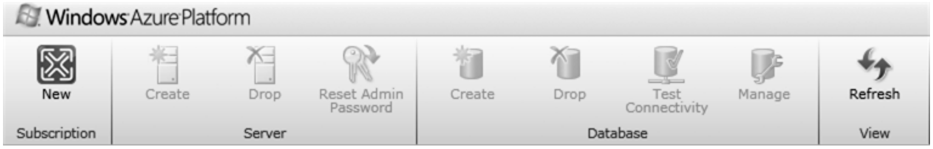


Figure 7-24. Management portal commands in the Database context

Figure 7-25 illustrates a typical developer workflow on the Windows Azure Platform.

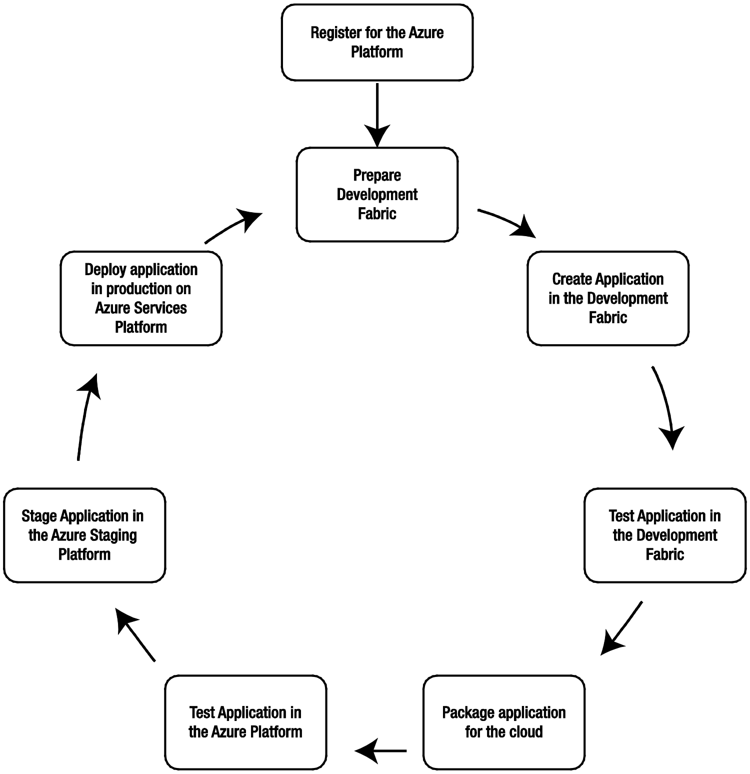

Register for the Azure Platform
Prepare Development Fabric
Create Application in the Development Fabric
Test Application in the Development Fabric
Package application for the cloud
Test Application in the Azure Platform
Stage Application in the Azure Staging Platform
Deploy application in production on Azure Services Platform

Figure 7-25. *Windows Azure platform developer workflow*

The typical developer workflow steps for Windows Azure platform follow:

1. Create a Windows Azure account (i.e., an account for Windows Azure, AppFabric, or SQL Services).
2. Download and prepare the development fabric to create a local cloud platform.
3. Create an application in the development fabric.
4. Test the application in the development fabric.
5. Package the application for cloud deployment.
6. Test the application on Windows Azure in the cloud.
7. Stage the application in the Windows Azure staging environment in the cloud.
8. Deploy the application in the production farm.

Windows Azure platform is a living platform; new features are added every few weeks. One such feature that I have not covered in detail is the Windows Azure Marketplace DataMarket. I will give you a brief overview of the topic on this subject that will conceptually understand the service offering.

Windows Azure Marketplace DataMarket

Windows Azure Marketplace DataMarket is a data service broker that runs on Windows Azure. It standardizes the data consumption and data publishing interfaces with a standard web protocol called Open Data Protocol (OData). Let's say you want to build a mobile application that gives end users insights into real estate sales and rental rates in relationship to local crime statistics. What would be your process for building such an application? The application is completely driven by public domain data and three different data sources: Real Estate Sales, Rental Data, and Crime Statistics. Then you realize that the programmatic interfaces for these data sources are different. Ultimately, you end up building your own service that transforms and aggregates the data from these three sources and presents it to the user.

The DataMarket standardizes this conversation by brokering such data feeds in OData format. OData is a web protocol that standardizes data publishing and consumption so that tightly integrated data can be exposed as standardized cross-platform feeds. You can find more information on OData at `www.odata.org`. The web site also provides client (consumption) and server (publisher). As an ISV or a developer, DataMarket provides you with a single point for consuming public domain as well as premium commercial data in a standard format. You maintain a single billing relationship with Microsoft and not worry about signing multiple checks for data providers. As a data provider, DataMarket provides you with a global brokerage service for monetizing your data by delivering it anywhere in the world. DataMarket also provides an authorization model for delivering your data only to your customers. You don't have to worry about maintaining separate customer relationships and separate infrastructure for publishing your data globally. You can find more information on the DataMarket on its web site `https://datamarket.azure.com/`. Figure 7-26 illustrates the high-level architecture of the DataMarket platform.

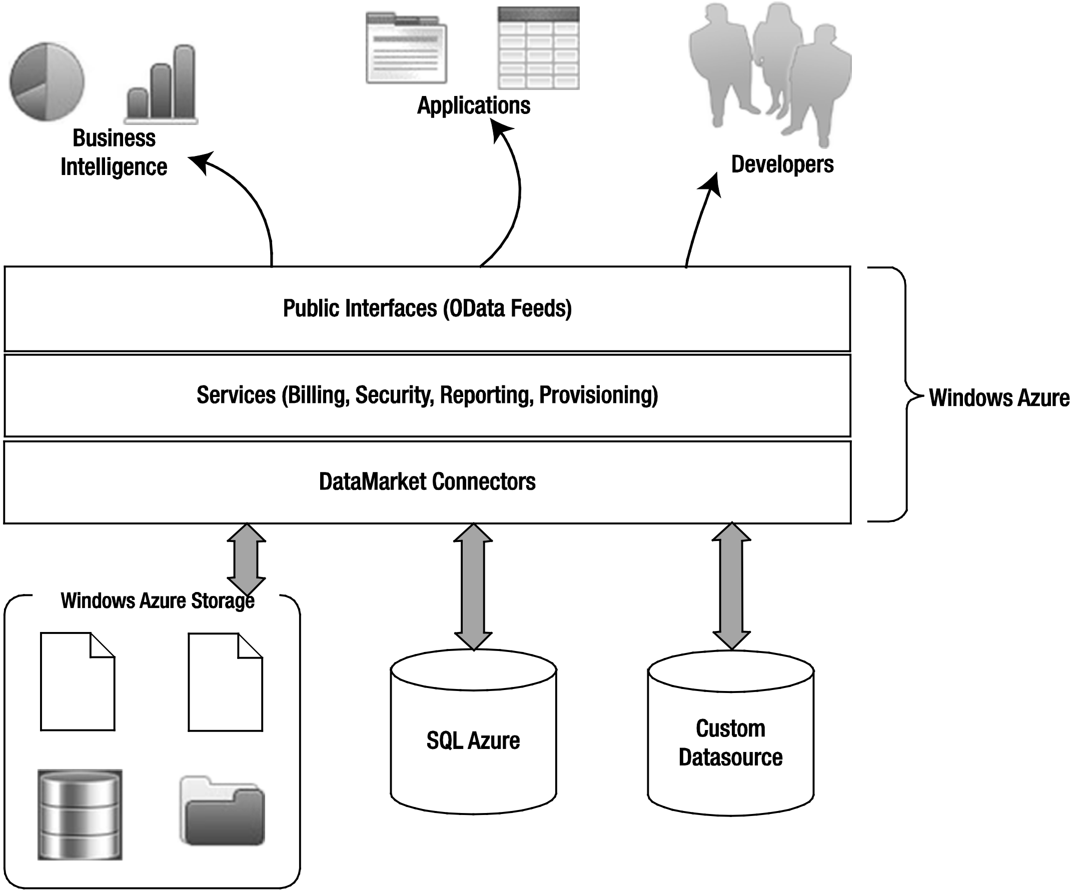


Figure 7-26. *Windows Azure Marketplace DataMarket*

As shown in Figure 7-26, you can expose your custom data source, SQL Azure data source, or a data source from Windows Azure Storage to the DataMarket. The DataMarket itself does not store your data, but provides you with a marketplace for your data. The DataMarket connectors provide the API services for publishing your data to the DataMarket. The DataMarket in turn exposes the data to the consumers as OData feeds. The services layer manages the billing, security, reporting, and provisioning.

Knowledge is best applied in the context of real-world scenarios. In the next section, you will see some common scenarios relevant to the Windows Azure platform components I have covered so far.

Windows Azure Platform Common Scenarios

After working with several customers over the past few years on the Windows Azure platform, I have compiled a list of most commonly used scenarios. I have grouped these scenarios into three primary categories: Foundational, Enterprise, and ISV.

Foundational Scenarios

Foundational scenarios are core scenarios commonly found in the architectures of the cloud applications. These scenarios leverage the core features of PaaS offerings. Table 7-4 lists the descriptions of these foundational scenarios.

Table 7-4. Foundational Scenarios

Foundational Scenarios	Description
Dynamic Scaling	This is a core capability of PaaS. Windows Azure platform offers this capability through APIs. In this scenario, a scaling engine keeps track of the performance of your instances and then based on a business rule like performance threshold or time, dynamically increases or decreases the number of instances in your cloud application.
Distributed Caching	Cloud applications run on shared hardware in Microsoft's data center. You do not have access to server and network hardware for optimizing performance at the hardware layer. Therefore, you have to optimize software for performance by employing techniques like distributed caching. Windows Azure AppFabric caching provides such a service that you can use in your cloud applications.
Multi-tenancy	For serving multiple customers from a single application, you have to implement multi-tenancy in all the tiers of your application. Every tier has its own design patterns for building multi-tenant applications. You can also deploy a separate application per tenant if your business model dictates that. Billing and metering per tenant is the biggest challenge in building multi-tenant applications. You need to provide data capture hooks within the application for capturing each tenant's usage of the application.
Geo-replication	With the spread of consumer and enterprise applications at a global scale, replicating data around the world and also employing bi-directional synchronizations for that data has become a necessity. SQL Azure DataSync and Microsoft Sync Framework are two technologies you can use to not only replicate data across the globe but also trickle down the data all the way to consumer devices for optimal performance and pleasant user experiences.
Identity Management	Identity management is one of the core requirements for building an extensible and backwards-compatible cloud application. Integrating enterprise and consumer identities seamlessly in your cloud application is important to make it pervasive and backwards compatible without modifying any code. Windows Azure AppFabric Access Control Service and Windows Identity Foundation (WIF) provide you with a service and framework for integrating different kinds of identity providers in your application.

Storage Management	Even though Windows Azure platform provides you with Storage service and SQL Azure, you have to architect security and the flow of data from data sources to cloud storage and back. You have to explicitly implement a storage management solution in your cloud applications.

Enterprise Scenarios

Enterprise scenarios are typically seen in enterprise cloud applications. Enterprises are business entities that manage a business and also interact with other businesses called partners. Enterprise scenarios are tightly bound to enterprise boundaries like enterprise identities and enterprise security policies. Table 7-5 lists the descriptions of these enterprise scenarios.

Table 7-5. Enterprise Scenarios

Enterprise Scenarios	Description
Enterprise Identity Integration	Enterprise cloud applications need access to enterprise identities for authentication and authorization. Windows Azure platform currently does not provide you with Active Directory services; therefore you have to leverage tools like ADFS 2.0 and WIF for integrating enterprise on-premises identities into your cloud applications. These tools also enable you to integrate seamlessly with identities of your business partners (partner portals, disaster recovery applications, and so forth).
Application Migrations	Application migration is a common scenario in enterprises. Windows Azure platform provides you with a runtime for running consolidated sets of applications. Enterprises build application consolidation strategies around Windows Azure platform in order to reap the benefits of cost, agility and high-availability.
Data Migrations	Many enterprises would like to relieve themselves of managing any non-sensitive data. Typically, data in enterprises reside in structured and unstructured sources. Enterprises leverage Blob storage and SQL Azure for storing their unstructured and relational data. Once the data is migrated and organized in the cloud, any application from anywhere can access this data securely.
Third-party Applications	Enterprises are seeing the VM role as an attractive option for deploying third-party applications in Windows Azure. Typically, these applications have a lengthy installation process and do not need to scale.
Business Intelligence (BI) in the Cloud	With unlimited storage and computing capacity in the cloud, enterprises have the opportunity to mine business data without worrying about the computing power needed for it. SQL Reporting Services in the cloud and Silverlight are popular BI presentation tools

	available in the Windows Azure platform. I commonly see departmental BI reporting applications using SQL Azure and Silverlight built on the Windows Azure platform. SQL Server Analysis Services (SSAS) and SQL Server Integration Services (SSIS) are still not available in the cloud to move the end-to-end BI process in the cloud, but I do see it coming in the next couple of years.
Hybrid Applications	Large enterprise applications like line-of-business applications are not yet available in the cloud, and enterprises have to settle for running them on-premises. There are also scenarios where 90% of the application can run comfortably in the cloud, but the other 10% requires access to sensitive data that cannot be moved to the cloud or the data resides in a line-of-business application. In this case, enterprises leverage either Windows Azure AppFabric Service Bus or Windows Azure Connect for retrieving the 10% data from the on-premises source.

ISV Scenarios

The Windows Azure platform is very attractive for ISVs because they can deploy their software service and offer it to multiple customers through different endpoints. They don't have to manage any hardware and can dynamically scale as the number of customers increases. Table 7-6 lists the descriptions of these scenarios.

Table 7-6. *ISV Scenarios*

ISV Scenarios	Description
High-Scale/Batch Compute	High-scale compute requires dynamically scaling compute power and the ability to turn the capacity off once the workload completes to avoid too much idle time. Similarly, any batch processing system only requires a specific timeslot for processing the load. The system usually runs idle till the next workload is initiated. Windows Azure platform provides you with unlimited compute power, dynamic scalability, and you can stop the system after the workload finishes so that you don't have to run any idle capacity.
High-Growth Sites	Startup companies or new initiatives in larger companies leverage the dynamic scaling capability of the Windows Azure platform by quickly building and deploying a new application scale up dynamically as the demand grows. If the demand goes down, the application can be easily scaled down or removed. There is no capital investment required upfront for designing the system for maximum capacity.
Software Modernization	There are a lot of ISV-packaged software installations that run as islands within enterprises and small businesses. With the wave of

	mobile applications, the demand for these packaged software applications is going down. Some ISVs are modernizing these software installations by providing Windows Azure AppFabric Service Bus interfaces and providing mobile access to these service interfaces.
Cloud Bursts (Predictable/ Un-predictable)	Bursts are sudden changes in the usage demand of an application. For example, during the Super Bowl, there is a sudden change in demand for pizza orders. If the system is not designed to handle these bursts, the company may lose business. Bursts can be predictable or unpredictable. In Windows Azure, you can detect these bursts by monitoring performance of the instances or input queues to the application. Then, based on business rules, you can increase the capacity of your application by dynamically starting more instances.

These categories provide some general guidance but are not specifically driven by either ISV or enterprise applications. In the real world, you will see a mix of these scenarios across different types of businesses.

Summary

Windows Azure platform is the most comprehensive PaaS offering in the industry today. In this chapter, I gave you a high-level overview of all the Windows Azure platform features. I also went over some of the common scenarios and trends I have observedg in the cloud computing industry. In the next few years, you will clearly see the disruptive effect of the cloud in enterprises. I always recommend that my clients have a top-down strategy in handling this disruption rather than on a per-application basis. As the platform matures in features, more and more enterprises will be moving their applications to the cloud.

In the next chapter, I will go over the Windows Azure Compute service in detail. You will also learn to build applications for the Windows Azure Compute service.

Bibliography

Apache Software Foundation. (n.d.). *Apache Hadoop*. Retrieved from `http://hadoop.apache.org`

Factor, A. (2001). *Analyzing Application Service Providers*. Prentice Hall.

Google. (n.d.). *Google AppEngine*. Retrieved from Google: `http://code.google.com/appengine`

Google. (n.d.). *Google Apps*. Retrieved from Google Apps: `http://www.google.com/apps/intl/en/business/index.html`

Mario Barbacci, M. H. (1995). *Quality Attributes*. Pittsburgh, Pennsylvania 15213: Software Engineering Institute, Carnegie Mellon University.

Microsoft Corporation. (n.d.). *About Windows Azure*. Retrieved from Windows Azure: `http://www.azure.com/`

Microsoft Corporation. (n.d.). *Windows Azure Pricing*. Retrieved from Windows Azure: `http://www.microsoft.com/azure/pricing.mspx`

Open ID Foundation. (n.d.). Retrieved from `http://openid.net/foundation/`

Staten, J. (2008). *Is Cloud Computing Ready For The Enterprise?* Forrester Research, Inc.

CHAPTER 8

Windows Azure Compute

Adapted from Chapter 2 of *Windows Azure Platform, 2/e*

Enterprises today run on several flavors of operating systems such as Windows, UNIX, and mainframes. As businesses grow, enterprises have to expand their data and processing capacities by buying more servers and operating systems to support the new capacity. Typically, businesses have to plan for growth well in advance to budget the expenses. The tipping point, where investments in new server systems may not justify the value they provide to the business, is not far away. This is because server systems are expensive to provision and maintain, and they become obsolete before they provide any return on investment (ROI) to the business. As a result, IT managers face constant struggle in justifying server upgrades. On the other hand, businesses should also plan for shrinking capacity and should be able to quickly reduce costs and expenses to compete better in the marketplace.

By adopting Windows Azure, businesses can outsource their infrastructure to Microsoft, and thus dynamically adjust to the real-time business needs.

In this chapter, I will discuss Windows Azure Compute architecture and service management components in detail.

Compute Service

In Chapter 7, you learned the high-level Compute service architecture. In this chapter, I will discuss more of the details of the Compute service.

Figure 8-1 illustrates the key components of the Windows Azure Compute service.

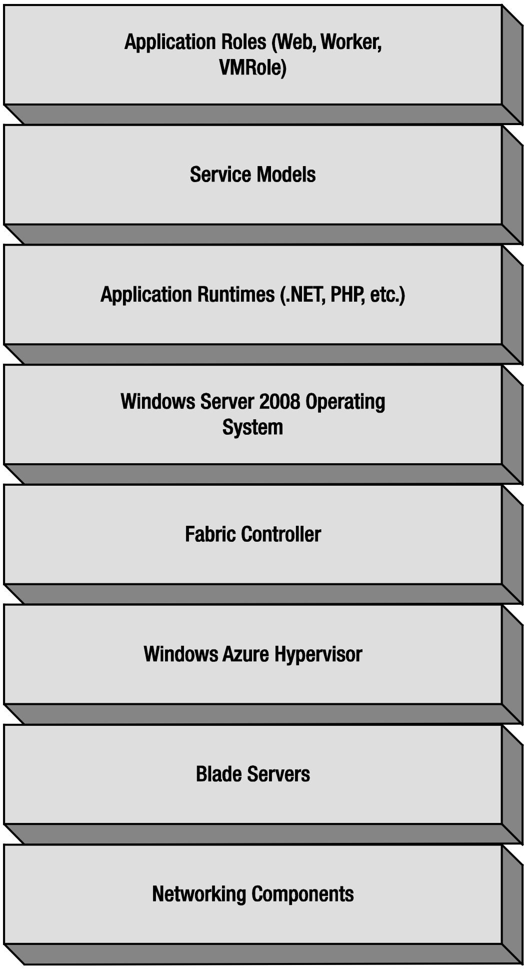

Application Roles (Web, Worker, VMRole)
Service Models
Application Runtimes (.NET, PHP, etc.)
Windows Server 2008 Operating System
Fabric Controller
Windows Azure Hypervisor
Blade Servers
Networking Components

Figure 8-1. *Windows Azure Compute service components*

Networking Components

The networking hardware and software forms the networking backbone for all the internal and external communications. Typical hardware components include gateways, routers, switches, hubs, and fiber optic cables across datacenters. Typical software networking components include software load-balancers and virtual routers. The hardware networking components are fixed, whereas the software networking components are dynamic and therefore can be provisioned on demand. From previous experience, the hardware networking components are more difficult to replace than software ones and therefore need to be redundant. One of the common mistakes is deploying two instances of the load-balancer (or router) from the same vendor as a redundancy mechanism. So, if there is a defect in the product version that freezes the device, the redundant device is also likely to fail, because it is identical. If the networking backbone fails, everything above the stack fails. Therefore, designing network redundancy with multi-vendor products is critical for maintaining high-availability of the data center services.

Blade Servers

The server farm in Windows Azure data centers consists of commodity-grade blade servers. The server hardware is low cost and is replaced instead of repaired in case of failures. This is because the replacement cost of the hardware is less than the repair cost. All the hardware is capable of running Windows Server 2008 64-bit compatible operating systems.

Windows Azure Hypervisor

The Windows Azure Hypervisor is a customized version of Hyper-V for Windows Azure. It is specifically tailored for Windows Azure.

Fabric Controller

The Fabric Controller is the heart of Windows Azure and is responsible for the following:

- Provisioning operating systems on server hardware.
- Allocating resources for cloud services.
- Managing cloud service life cycle.
- Maintaining the cloud service quality attributes defined by the service level agreement (SLA).

Operating Systems

Table 8-1 shows the versions of guest operating systems supported by Windows Azure.

Table 8-1. *Windows Azure Operating System Support*

Windows Azure Operating System	Windows Server OS Version	Roles
Guest OS 1.x	Windows Server 2008 (64 bit)	Web role, Worker role
Guest OS 2.x	Windows Server 2008 R2 (64 bit)	Web role, Worker role, VM role

Application Runtimes

Application runtimes, like .NET Framework and Java Runtime, are responsible for running your applications on Windows Azure. By default, .NET Framework is installed on these instances, but you can also install Java runtime or any runtime that runs on Windows Server 2008.

Service Model

A service model contains the metadata for your service. It describes the behavior of the service after it is deployed in Windows Azure. The Fabric Controller reads the service model definition of your service and deploys it to the desired end state. You can modify the configuration portion of the service model, such as the number of instances at runtime. Any modification to the definition portion of the service mode like the number of endpoints requires a restart. The service model must contain at least one role definition.

Application Roles

Application roles abstract the dependency of specific operating features from your application. For example, if you build a Web role, the web server in which the Web role is hosted is abstracted from you. The underlying web server in Windows Azure is IIS, but your application does not have to know about it. Windows Azure deploys your Web role to an IIS server transparently, thus relieving you of web application deployment task. Similarly, a Worker role is deployed as a background service in Windows Azure without requiring you to specifically install and deploy the service on each individual instance. The Compute service follows a role-based model in which the provisioned instances run the roles defined by your application.

■ **Note** In Windows Azure terminology, an application is called as a cloud service. A cloud service is a grouping of one or more roles into a single packaged solution that can be deployed to Windows Azure as a distributed application farm.

Windows Azure supports three types of roles: Web role, Worker role, and VM role. When you design your cloud service, you define your architecture in terms of roles. Then, you group multiple roles as a cloud service and deploy to Windows Azure. Windows Azure deploys each of these roles into a separate virtual machine instance but within the same cloud service. Figure 8-2 illustrates the concept of cloud service and roles.

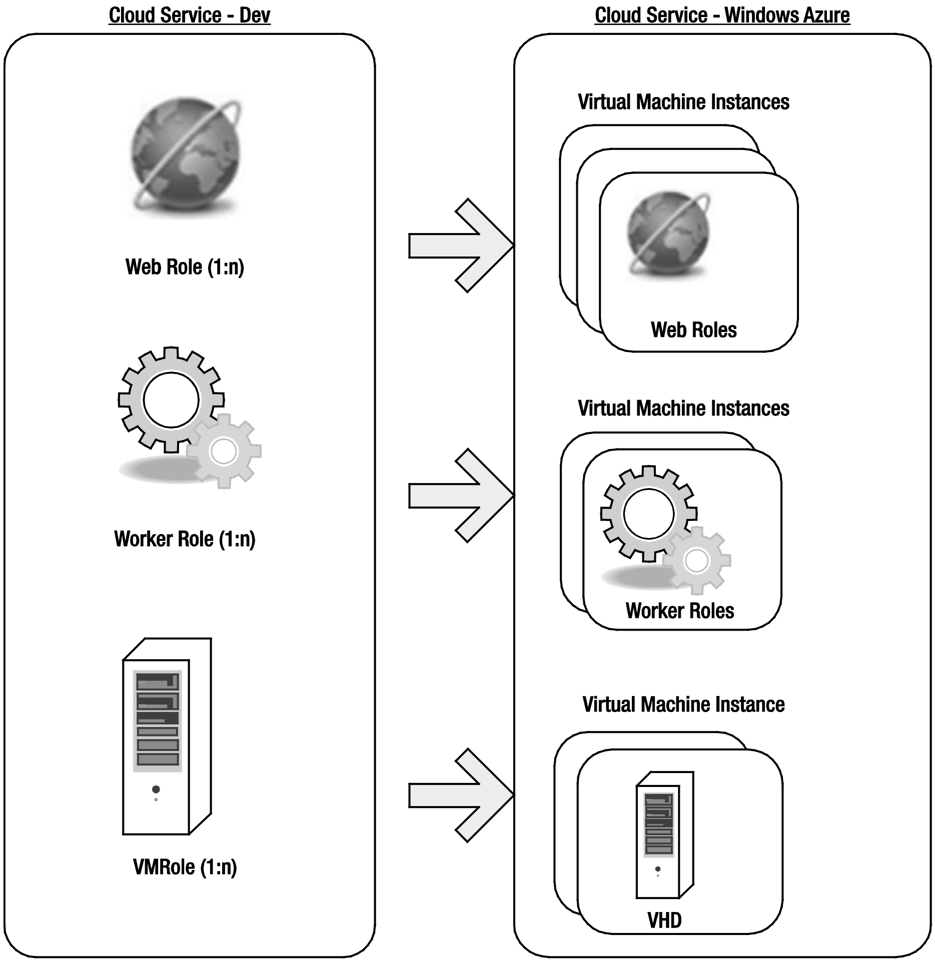


Figure 8-2. Windows Azure roles and cloud service

Each role can be deployed to Windows Azure as multiple instances, and can also be scaled up and down dynamically via the Service Management API.

Upgrade Domains and Fault Domains

The Service Level Agreement (SLA) for Windows Azure states, "For Compute, we guarantee that when you deploy two or more role instances in **different fault and upgrade domains** your Internet facing roles will have external connectivity at least 99.95% of the time."[1]

One of the value propositions of Windows Azure is the ability to automatically update instances in your service. But, if the update requires a server reboot, it directly affects the availability of your service.

Upgrade Domains are a logical separation of your role instances, ensuring your service does not go down during upgrades. There are five upgrade domains by default, but you have control over the number of upgrade domains your application will use through the service model definition.

Fault domains are logical separations of your role instances for avoiding single point of failure due to software and hardware failures. The Fabric Controller always deploys more than one instances of your service in two separate fault domains. Therefore, the SLA specifically requires you to have at least two instances of a role for 99.95% availability. The SLA also states, "Additionally, we will monitor all of your individual role instances and guarantee that 99.9% of the time we will detect when a role instance's process is not running and initiate corrective action." You don't have any control over the number of Fault Domains and the assignment process. The Fabric Controller internally defines a unit of failure and never deploys two instances of the same role in the same unit of failure. Therefore, if there is a hardware failure in a rack on which one instance is running, the second instance still remains available. After detecting failure, the Fabric Controller then restarts a new instance in a third fault domain. All this happens automatically without any user intervention. If there are three instances of the role: the Fabric Controller decides which Fault Domain to place these instances. It may even place two instances in the same Fault Domain and the third in another. This is not an ideal configuration, but it may occur. By default, your service gets two Fault Domains, therefore the likelihood of the first and the third instance running in the same Fault Domain is high. Windows Azure SDK provides you with a method call to get the Upgrade Domain and the Fault Domain of the role instance. I have covered this later in the chapter. The Fabric Controller also makes sure that one Upgrade Domain spans more than one Fault Domain as shown in Figure 8-3.

[1] Windows Azure SLA http://www.microsoft.com/windowsazure/sla/

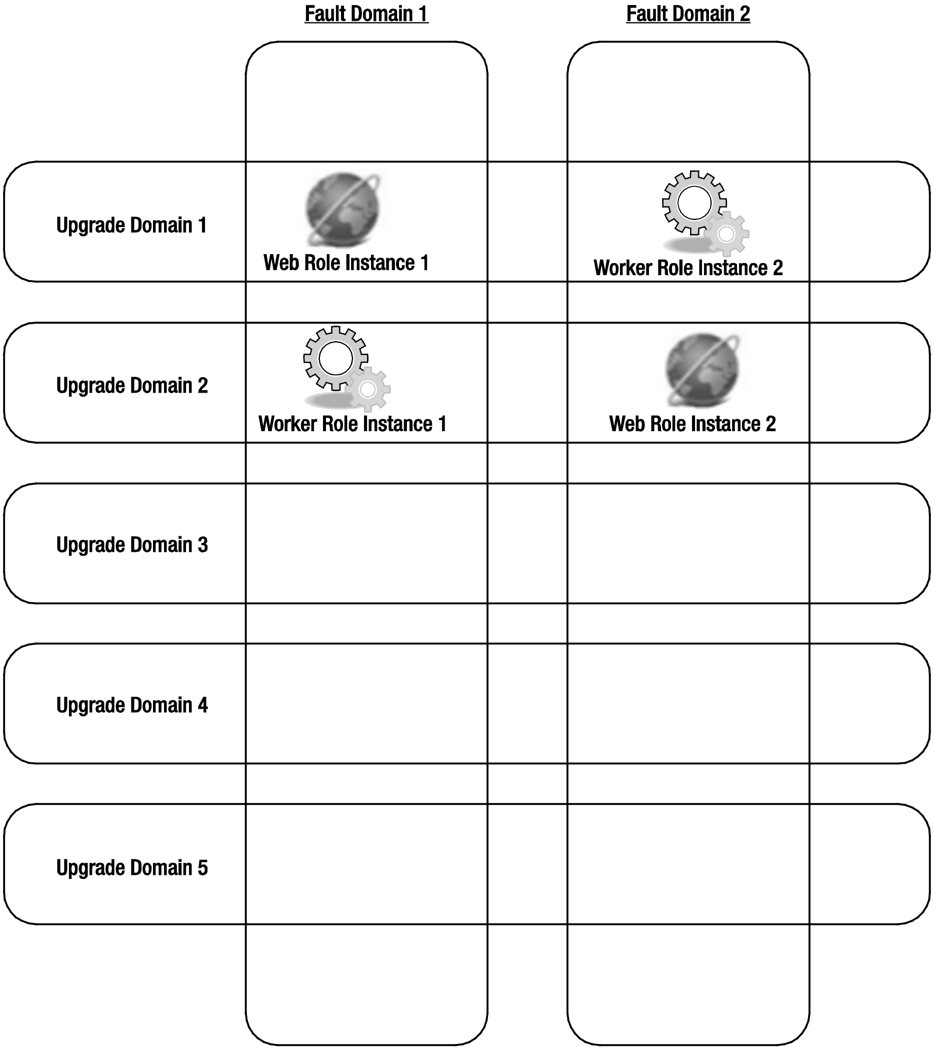


Figure 8-3. *Upgrade Domain and Fault Domain*

Understanding Domains in the Application Context

To further understand the concept of Upgrade and Fault Domains, let's take an example of a Windows Azure cloud service named "MyFirstService" that has only one web role, "MyWebRole." Initially, let's assume that MyWebRole has two instances, and that you are using two Upgrade Domains. The Fabric Controller will deploy each instance of MyWebRole into different Upgrade Domains and different Fault Domains as shown in Figure 8-4.

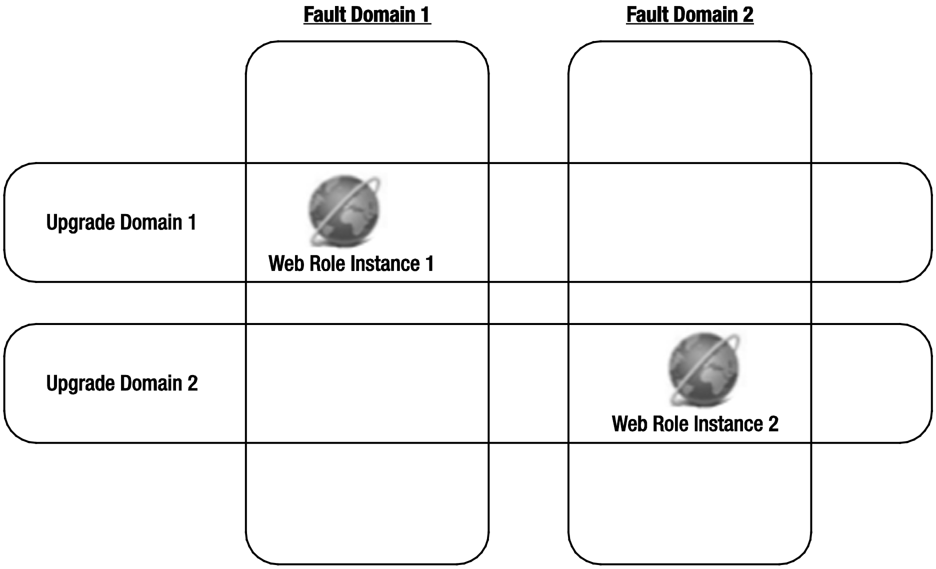


Figure 8-4. *Two Web role instances in Upgrade and Fault Domains*

Now, if you increase instances to three and add one more upgrade domain, then even though the Fabric Controller may deploy three instances in three different Upgrade Domains, there is no guarantee that the third instance will be deployed in a different Fault Domain. It may deploy the third instance to the same Fault Domain as the second instance, as shown in Figure 8-5.

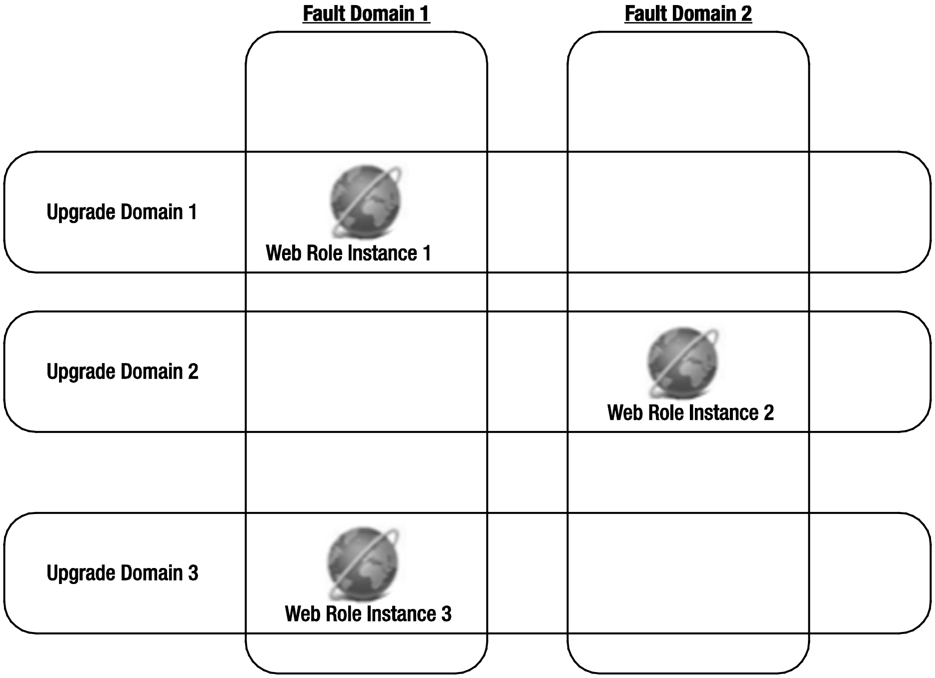


Figure 8-5. Three Web role instances in Upgrade and Fault Domain

While designing highly available services, it is important to understand the functioning of Upgrade and Fault Domains and to deploy the services with appropriate numbers of Upgrade domains for avoiding any downtime. Windows Azure SDK provides you with method calls to find out the Upgrade and Fault Domain numbers in which a role instance is running. I cover these later in this chapter.

Compute Service Security

When you run services in someone else's datacenter, there is always a concern about the security of application and data. In Compute, the instances of your service run in dedicated virtual machines. These virtual machines are managed by the hypervisor. Your deployed service runs as a least privileged account in the dedicated virtual machine. The root virtual machine is trusted by the hypervisor, but the guest virtual machine, in which your service runs, is not trusted. The root virtual machine and the hypervisor also have network packet filters that prevent unauthorized traffic to the virtual machines running the service. The isolation of virtual machines from one another is managed by the hypervisor and does not depend on Windows security. For avoiding side channel attacks, each virtual machine is isolated into a separate core. Figure 8-6 illustrates the virtual machine isolation and some security features of the Compute service.

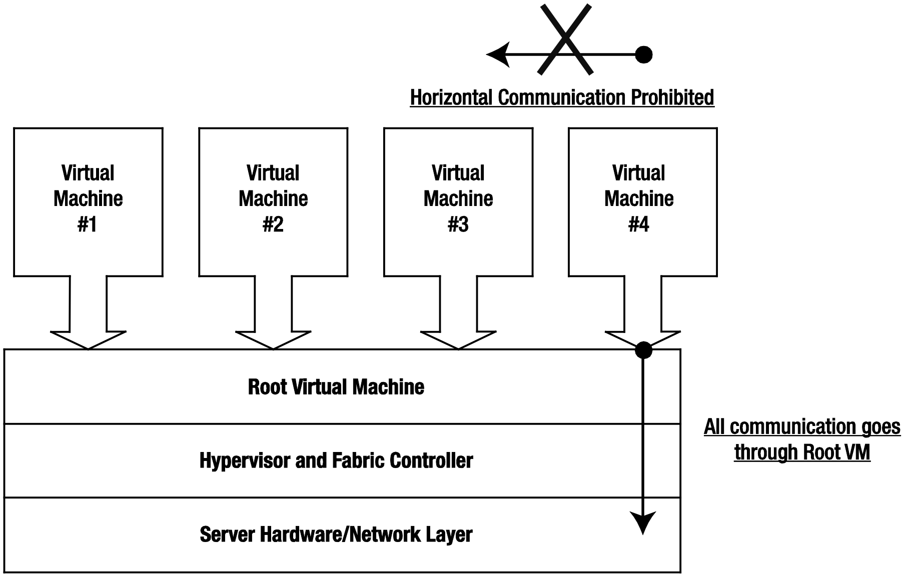

Horizontal Communication Prohibited
Virtual Machine #1
Virtual Machine #2
Virtual Machine #3
Virtual Machine #4
Root Virtual Machine
Hypervisor and Fabric Controller
Server Hardware/Network Layer
All communication goes through Root VM

Figure 8-6. *Compute service security.*

The commonly exposed attack surface for a Web role is the external endpoint. Therefore, it is important to secure the external endpoints exposed by your service using secure channels like https or transport/message security in case of web services. You still have to manage any denial of service attacks in your application if they are not detected by the Windows Azure platform infrastructure.

Developing Windows Azure Services

Windows Azure and Windows Azure AppFabric have separate software development kits (SDKs), but Visual Studio and the .NET Framework are the common programming tools used for building applications for all the Windows Azure components.

■ **Note** If you are a Java or a PHP developer, you can also build services in Windows Azure. Check out the following web sites:

www.interoperabilitybridges.com/projects/windows-azure-tools-for-eclipse
www.interoperabilitybridges.com/projects/windows-azure-sdk-for-java
www.windowsazure4j.org

Windows Azure SDK has a local development fabric that simulates the cloud environment at a miniature scale. Developers can utilize their existing .NET development skills for developing services for

Windows Azure platform. The development fabric gives you a local environment to test your code. The development fabric is not meant for running any production applications.

Windows Azure API Structure

Windows Azure SDK provides a set of APIs to complement the core services offered by Windows Azure. These APIs are installed as part of Windows Azure SDK, and can be used locally for developing Windows Azure applications. The `Microsoft.WindowsAzure.ServiceRuntime` assembly and namespace consists of classes used for developing applications in the Compute service. The `Microsoft.WindowsAzure.Diagnostics` namespace consists of classes used for diagnostics and logging in the Compute service.

The `Microsoft.WindowsAzure.StorageClient` assembly and namespace consists of classes used for developing applications to interact with the Storage service. The assembly makes REST calls to the Storage service REST interface.

The Service Management API is exposed as a REST interface, and the `csmanage.exe` application in Windows Azure code samples (`http://code.msdn.microsoft.com/windowsazuresamples`) can be used to call the Service Management APIs.

Developer Environment

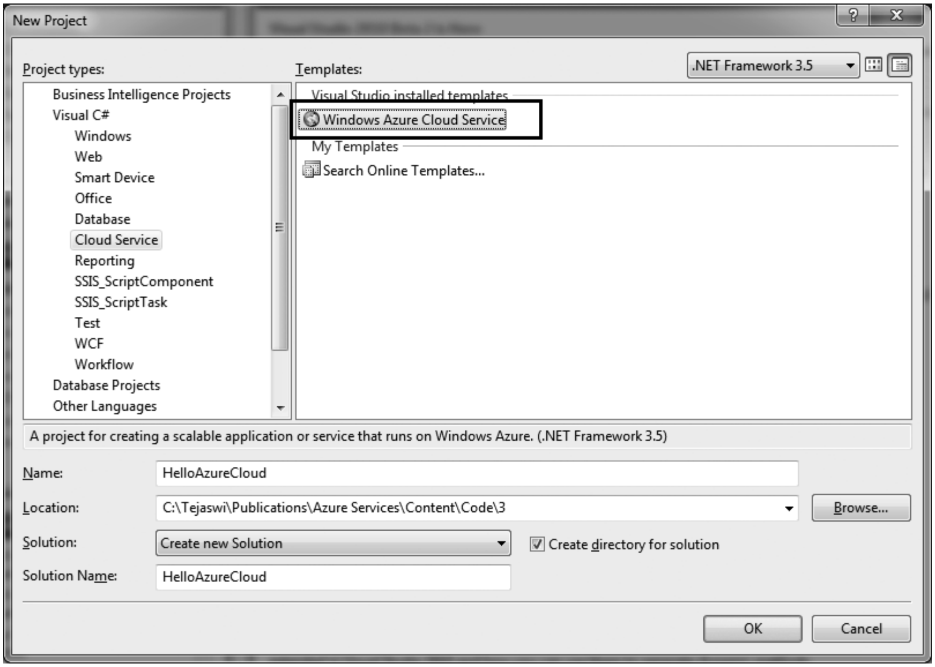
The development environment of Windows Azure consists of two main components: Windows Azure Tools for Visual Studio and the Windows Azure SDK. In this section, I will cover these in detail.

Windows Azure Tools for Visual Studio

Windows Azure Tools for Visual Studio is a Visual Studio extension supporting Windows Azure development. You can download it from the Azure SDK web site at `www.microsoft.com/azure/sdk.mspx`.

Visual Studio Project Types

The Windows Azure Tools for Visual Studio creates a project type named Cloud Service containing project templates for Web role and Worker role. After you install Windows Azure Tools for Visual Studio, open Visual Studio and create a new Project by selecting File New Project. Figure 8-7 shows the New Project dialog box.



New Project

Project types:

- Business Intelligence Projects
- Visual C#
  - Windows
  - Web
  - Smart Device
  - Office
  - Database
  - Cloud Service
  - Reporting
  - SSIS_ScriptComponent
  - SSIS_ScriptTask
  - Test
  - WCF
  - Workflow
- Database Projects
- Other Languages

Templates: .NET Framework 3.5

Visual Studio installed templates

- Windows Azure Cloud Service

My Templates

- Search Online Templates...

A project for creating a scalable application or service that runs on Windows Azure. (.NET Framework 3.5)

Name: HelloAzureCloud

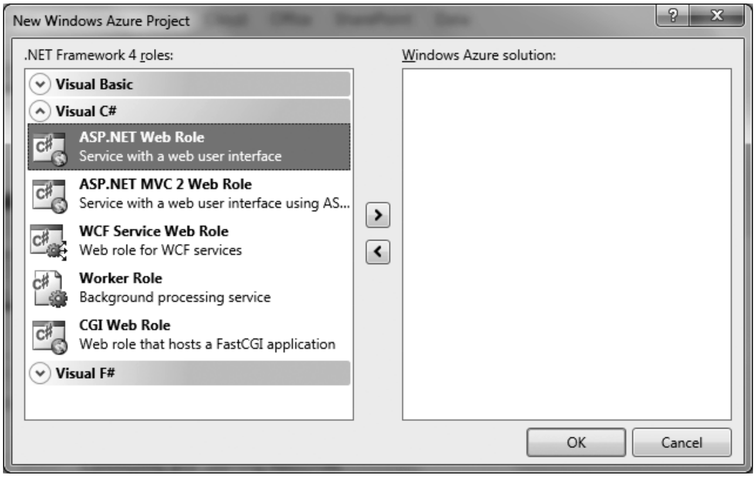
Location: C:\Tejaswi\Publications\Azure Services\Content\Code\3

Solution: Create new Solution ☑ Create directory for solution

Solution Name: HelloAzureCloud

Figure 8-7. *New project*

The Windows Azure Cloud Service template defines the cloud service project. Click OK to choose from the available roles (see Figure 8-8).



New Windows Azure Project

.NET Framework 4 roles:

- Visual Basic
- Visual C#
  - **ASP.NET Web Role** – Service with a web user interface
  - **ASP.NET MVC 2 Web Role** – Service with a web user interface using AS...
  - **WCF Service Web Role** – Web role for WCF services
  - **Worker Role** – Background processing service
  - **CGI Web Role** – Web role that hosts a FastCGI application
- Visual F#

Windows Azure solution:

Figure 8-8. *Cloud service roles*

The available cloud service roles are as follows:

- *ASP.NET and ASP.NET MVC Web role*: As the name suggests, this role consists of an ASP.NET project. You can build any ASP.NET compatible project for deploying to the cloud.
- *WCF Service Web role*: This role consists of a WCF project with basic HTTP binding.
- *Worker role*: The Worker role project is a background process application. It is analogous to a Windows service. A Worker role has start and stop methods in its super class and can expose internal and external endpoints for direct access.
- *CGI Web role*: The CGI Web role is a FastCGI-enabled Web role. It does not consist of a Cloud Service project.

Choose the roles you want, as shown in Figure 8-9. I have selected a Web role, a WCF Web role, and a Worker role.

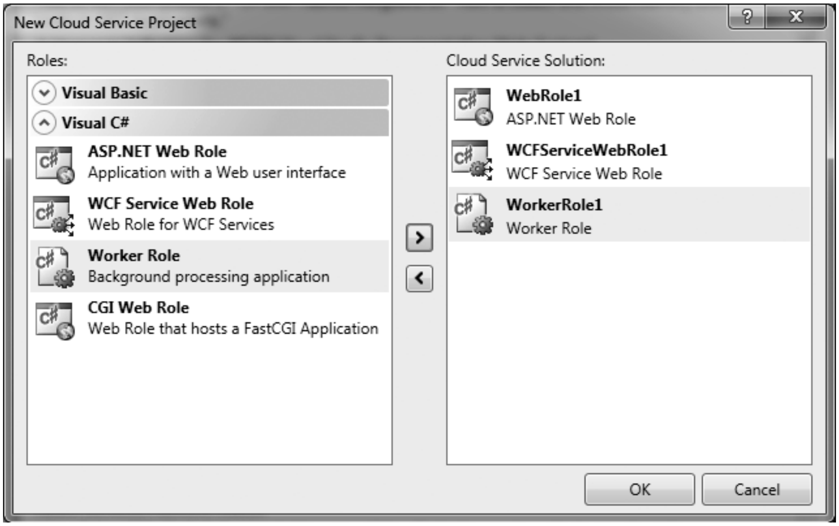
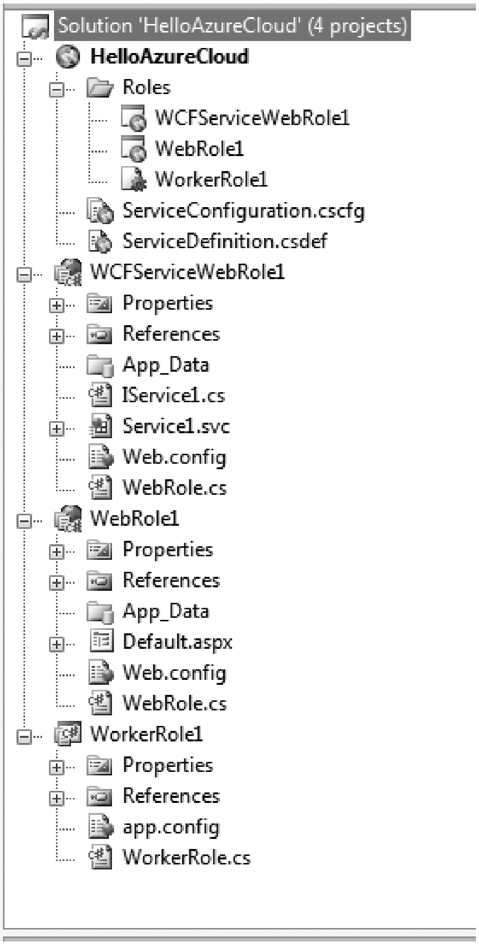


Figure 8-9. *Selected roles*

Click OK to create the cloud service project, as shown in Figure 8-10.



- Solution 'HelloAzureCloud' (4 projects)
  - HelloAzureCloud
    - Roles
      - WCFServiceWebRole1
      - WebRole1
      - WorkerRole1
    - ServiceConfiguration.cscfg
    - ServiceDefinition.csdef
  - WCFServiceWebRole1
    - Properties
    - References
    - App_Data
    - IService1.cs
    - Service1.svc
    - Web.config
    - WebRole.cs
  - WebRole1
    - Properties
    - References
    - App_Data
    - Default.aspx
    - Web.config
    - WebRole.cs
  - WorkerRole1
    - Properties
    - References
    - app.config
    - WorkerRole.cs

Figure 8-10. *Empty cloud service project*

In Figure 8-10, the `HelloAzureCloud` cloud service project holds references in the `Roles` subfolder to all the role projects in the solution. The cloud service project also contains `ServiceDefinition.csdef` and `ServiceConfiguration.cscfg` files that define the configuration settings for all the roles in the cloud service.

The WCF service Web role project includes a sample service and its associated configuration in the `web.config` file. The `WebRole.cs` file implements the start and configuration changing events fired by the Windows Azure platform. This file is created for all the roles with default start and configuration changing event handlers. You can handle additional events like `StatusCheck` and `Stopping` depending on your application needs. The `WebRole` class inherits the `RoleEntryPoint` class from the `Microsoft.WindowsAzure.ServiceRuntime` namespace. The `WebRole.cs` class is the background class that instantiates a background processing object at runtime. The Web role does not need to have this class, but it is beneficial for performing background and startup tasks within the Web role.

The ASP.NET Web role project consists of a `Default.aspx` file and its associated code-behind and `web.config` file.

■ **Caution** Even though Windows Azure supports native code execution, the code still runs with Windows user, not administrator, privileges. Therefore, some WIN32 APIs that require system administrator privileges will not be accessible

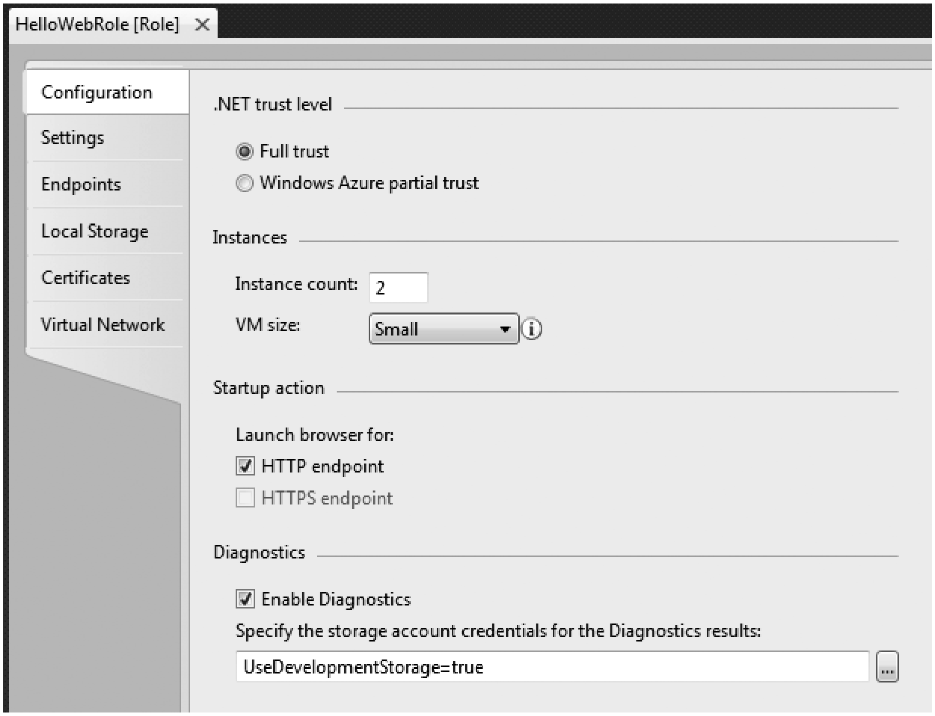
---

Finally, the Worker role project consists of `WorkerRole.cs` file and its associated `app.config` file. In addition to inheriting the `RoleEntryPoint` class, it also overrides the `Run()` method in which you add your continuous processing logic. A Worker role class must inherit from the `Microsoft.WindowsAzure.ServiceRuntime. RoleEntryPoint` class. `RoleEntryPoint` is an abstract class that defines functions for initializing, starting and stopping the Worker role service. A Worker role can stop either when it is redeployed to another server or when you have executed the Stop action from the Windows Azure developer portal. Because a Worker role is not designed to have any external interface by default, it does not contain any ASP.NET or WCF files.

In summary, the cloud service defined in this project consists of a WCF service, an ASP.NET web application, and a Worker role service. The entire package constitutes a Windows Azure cloud service.

Role Settings and Configuration

In the cloud service project, you can configure each role's settings by double-clicking the role reference in the `Roles` subdirectory of the cloud service project. Figure 8-11 shows the role settings page in Visual Studio.



.NET trust level

● Full trust
○ Windows Azure partial trust

Instances

Instance count: 2
VM size: Small

Startup action

Launch browser for:
☑ HTTP endpoint
□ HTTPS endpoint

Diagnostics

☑ Enable Diagnostics
Specify the storage account credentials for the Diagnostics results:
UseDevelopmentStorage=true

***Figure 8-11.** Role settings (the default is Configuration)*

■ **Note** The Role Settings UI actually updates the `ServiceDefinition.csdef` and `ServiceConfiguration.cscfg` files behind the scenes. You can achieve the same effect by directly modifying these XML files. The concept is important if you want to dynamically modify these files for auto-scaling of auto-deployments.

The role settings page has six tabs: Configuration, Settings, Endpoints, Local Storage, Certificates, and Virtual Network.

- *Configuration*: The Configuration tab is selected by default and displays the following configuration options:
- *.NET Trust Level*: The .NET Trust Level specifies the trust level under which this particular role runs. The two options are Full Trust and Windows Azure Partial Trust. The Full Trust option gives the role privileges to access certain machine resources and execute native code. Even in Full Trust, the role still runs in the standard Windows Azure user's context and not the administrator's context. In the Partial Trust option, the role runs in a partially trusted environment and does not have privileges for accessing machine resources and native code execution.

- *Instances*: The instance count defines the number of instances of each role you want to run in the cloud. For example, you can run two instances of ASP.NET Web role and one instance of the Worker role for background processing. The two instances of ASP.NET Web role will give you automatic load-balancing across the instances. By default, all the roles run as single instance. This option gives you the ability to scale-up and scale-down your role instances on demand.

The VM size option gives you the ability to choose from a list of virtual machines preconfigured in the Windows Azure virtual machine pool. You can choose from the following list of predefined virtual machines depending on your deployment needs:

- *Extra small*: 1x1.0 GHz core processor, 768GB RAM, 20GB hard disk (low IO)
- *Small*: 1x1.6 GHz core processor, 1.75GB RAM, 225GB hard disk (moderate IO)
- *Medium*: 2 core processors, 3.5GB RAM, 500GB hard disk (high IO)
- *Large*: 4 core processors, 7GB RAM, 1000GB hard disk (high IO)
- *Extra large*: 8 core processors, 15GB RAM, 2000GB hard disk (high IO)

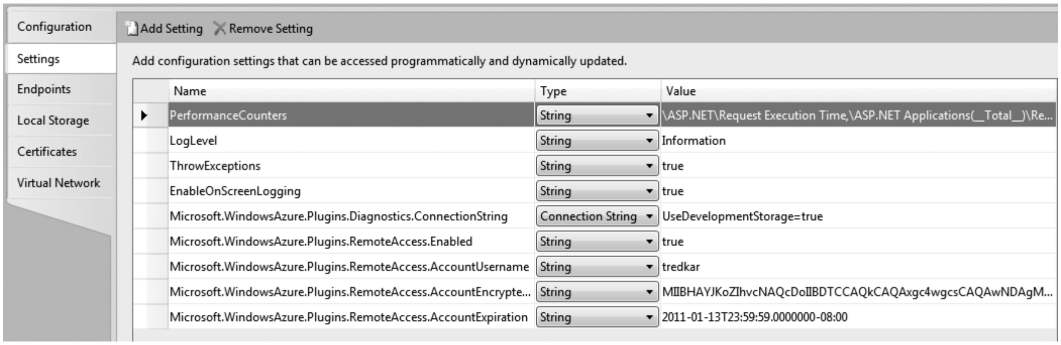
■ **Note** For any operations that require moderate to high I/O, I recommend using VM sizes of medium or higher. The I/O may include disk I/O, network I/O or even memory I/O like caching. I have also seen specific situations where 16 small VMs performed better than 2 extra large instances.

The Web roles have a startup action that defines the endpoint on which the browser should launch. This is not a cloud service setting, but a project setting for launching the Web role in the development fabric. The HTTPS checkbox will be disabled until you add an HTTPS endpoint to the role.

In the Diagnostics section, you can enable or disable the diagnostics capabilities for the role and also specify the destination of the diagnostics logs. By default, the destination points to the development storage: "`UseDevelopmentStorage=true`." If you have a Windows Azure Storage service created, you can add the connection string pointing to that storage service. The format for the storage service connection string is

```
DefaultEndpointsProtocol=https;AccountName=[Your Storage service account name];AccountKey=[You
can get this from the Management portal]
```

- *Settings*: The Settings tab, shown in Figure 8-12, defines any custom settings you can add to the role configuration.



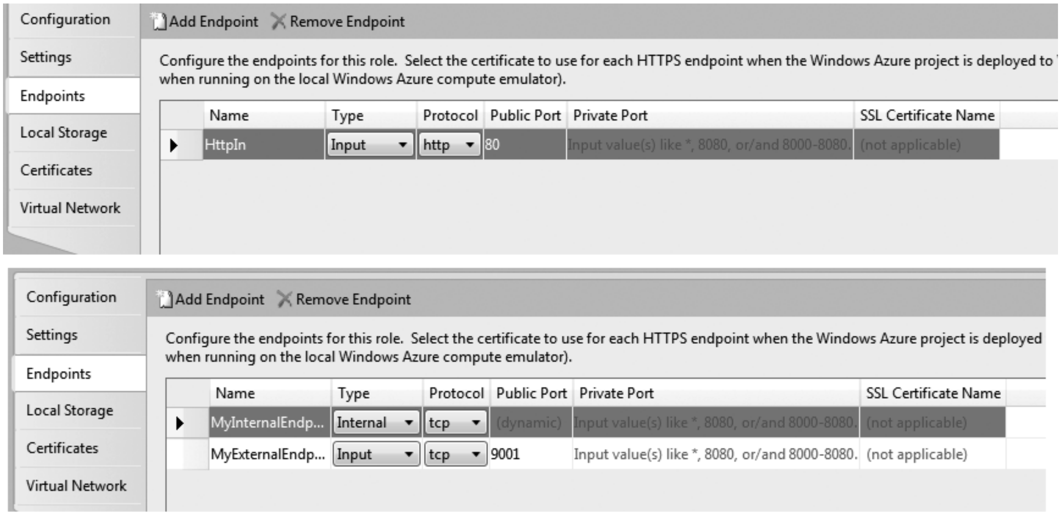
Add configuration settings that can be accessed programmatically and dynamically updated.

Name	Type	Value
PerformanceCounters	String	\ASP.NET\Request Execution Time,\ASP.NET Applications(__Total__)\Re...
LogLevel	String	Information
ThrowExceptions	String	true
EnableOnScreenLogging	String	true
Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString	Connection String	UseDevelopmentStorage=true
Microsoft.WindowsAzure.Plugins.RemoteAccess.Enabled	String	true
Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountUsername	String	tredkar
Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountEncrypte...	String	MIIBHAYJKoZIhvcNAQcDoIIBDTCCAQkCAQAxgc4wgcsCAQAwNDAgM...
Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountExpiration	String	2011-01-13T23:59:59.0000000-08:00

Figure 8-12. *Settings tab*

These custom name-value pairs are analogous to the name-value `appSettings` in an `app.config` or `web.config` file. You can retrieve the values of these settings in your code by calling the `RoleEnvironment.GetConfigurationSettingValue`. If you enable diagnostics or remote desktop access plug-ins, custom settings are created for these plug-ins in the Settings section. You can add your custom settings like the database connection string to this section.

- *Endpoints*: The Endpoints tab contains endpoints your role will expose when it is deployed. There is a hard limit of five endpoints you can expose from a role. Figure 8-13 shows the Endpoints tab for a Web role and a Worker role respectively.



Configure the endpoints for this role. Select the certificate to use for each HTTPS endpoint when the Windows Azure project is deployed to when running on the local Windows Azure compute emulator).

Name	Type	Protocol	Public Port	Private Port	SSL Certificate Name
HttpIn	Input	http	80	Input value(s) like *, 8080, or/and 8000-8080.	(not applicable)

Configure the endpoints for this role. Select the certificate to use for each HTTPS endpoint when the Windows Azure project is deployed when running on the local Windows Azure compute emulator).

Name	Type	Protocol	Public Port	Private Port	SSL Certificate Name
MyInternalEndp...	Internal	tcp	(dynamic)	Input value(s) like *, 8080, or/and 8000-8080.	(not applicable)
MyExternalEndp...	Input	tcp	9001	Input value(s) like *, 8080, or/and 8000-8080.	(not applicable)

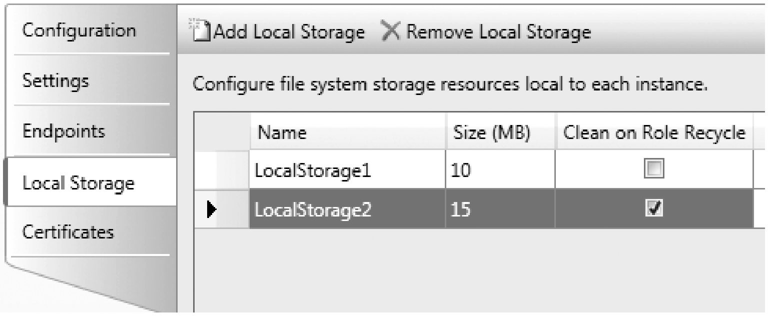
Figure 8-13. *Endpoints tab*

The roles can have `Input` endpoints and an `internal` endpoint. `Input` endpoints are exposed externally, whereas the internal endpoints are exposed internally within the cloud service role instances. The port number defines the port your will use while accessing the endpoint. In case of an HTTPS endpoint, you can upload the X.509 certificate for accessing the web page or service using an HTTPS encrypted connection.

The internal endpoint is accessible to other roles within the cloud service. For example, a Web role can get a reference to the internal endpoint of a worker role in the same cloud service for making web service method calls to it.

By default, a Worker role has no defined endpoints like a Web role, because it is intended to be used as a background process. To define an endpoint, you have to add one to the list and select its type (`input` or `internal`), protocol (tcp, http, and, https), port, and, optionally, an SSL certificate name.

- *Local Storage*: The Local Storage tab defines local directories that will be created on the server machine of the role for storing files locally. Figure 8-14 shows the settings on the Local Storage tab.



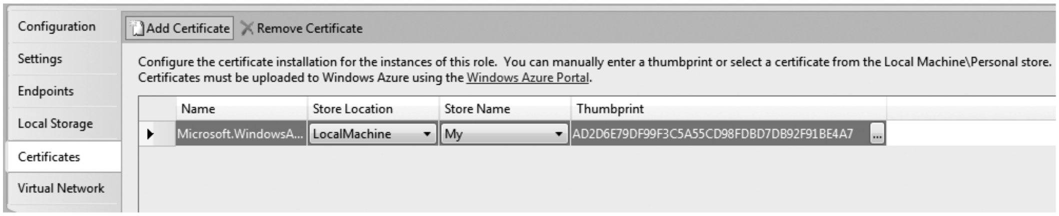
Configure file system storage resources local to each instance.

Name	Size (MB)	Clean on Role Recycle
LocalStorage1	10	□
LocalStorage2	15	☑

Figure 8-14. Local storage

The name of the local storage will be the names of directories created on the server. The size column defines the maximum size of the folder contents and the “Clean on Role Recycle” column defines whether you want the contents of the directory cleaned up when a role recycles. You can use this option for creating sticky storage for maintaining state of the role across reboots and failures. The local storage uses the overall hard disk capacity available for the role instance and therefore you have to make sure you don’t exceed the maximum capacity available for the virtual machine to avoid any errors.

- *Certificates*: The Certificates tab is used for referencing the certificates in your role. You can use the certificates referenced here for configuring HTTPS in your Web role and also for remote desktop connections and virtual network. The certificates listed here can be referenced throughout the role configuration. At the time of this writing, you still had to use the Windows Azure Management Portal or the Service Management API for uploading the certificate to the server and then reference the certificate in the settings as shown in Figure 8-15.

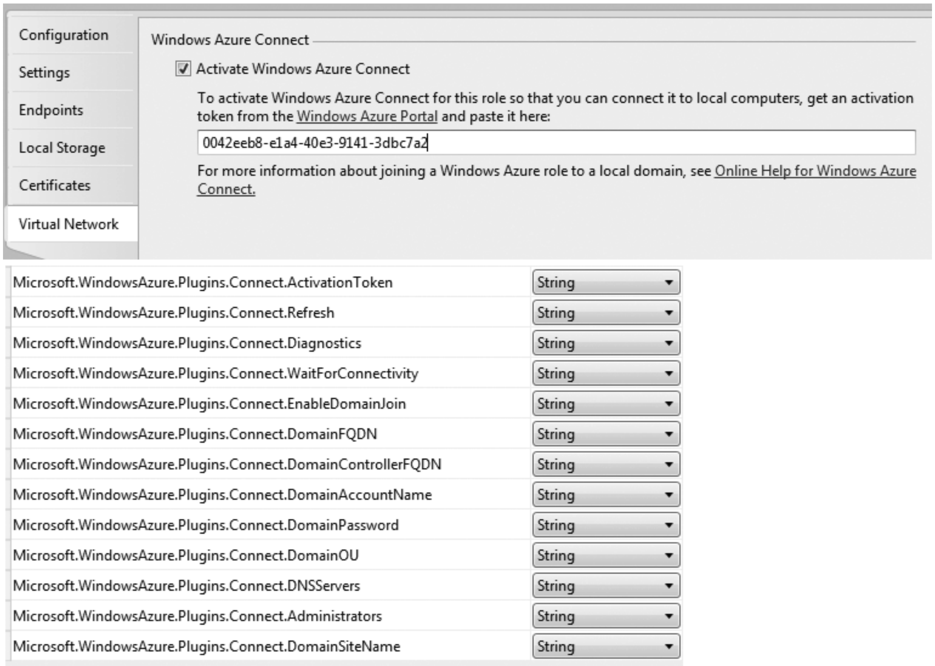


Configure the certificate installation for the instances of this role. You can manually enter a thumbprint or select a certificate from the Local Machine\Personal store. Certificates must be uploaded to Windows Azure using the Windows Azure Portal.

	Name	Store Location	Store Name	Thumbprint
▶	Microsoft.WindowsA...	LocalMachine	My	AD2D6E79DF99F3C5A55CD98FDBD7DB92F91BE4A7

Figure 8-15. *Certificate configuration*

- *Virtual Network*: The Virtual Network tab allows you create a virtual network between the Windows Azure role instances and your on-premises servers by adding a Windows Azure Connect activation token. You acquire the token from the Windows Azure Management Portal and copy it in this section. Once you activate the Virtual Network, Visual Studio creates several settings elements in `ServiceConfiguration.cscfg` for configuring the network, as shown in Figure 8-16.



Windows Azure Connect

☑ Activate Windows Azure Connect

To activate Windows Azure Connect for this role so that you can connect it to local computers, get an activation token from the Windows Azure Portal and paste it here:

0042eeb8-e1a4-40e3-9141-3dbc7a2

For more information about joining a Windows Azure role to a local domain, see Online Help for Windows Azure Connect.

Microsoft.WindowsAzure.Plugins.Connect.ActivationToken	String
Microsoft.WindowsAzure.Plugins.Connect.Refresh	String
Microsoft.WindowsAzure.Plugins.Connect.Diagnostics	String
Microsoft.WindowsAzure.Plugins.Connect.WaitForConnectivity	String
Microsoft.WindowsAzure.Plugins.Connect.EnableDomainJoin	String
Microsoft.WindowsAzure.Plugins.Connect.DomainFQDN	String
Microsoft.WindowsAzure.Plugins.Connect.DomainControllerFQDN	String
Microsoft.WindowsAzure.Plugins.Connect.DomainAccountName	String
Microsoft.WindowsAzure.Plugins.Connect.DomainPassword	String
Microsoft.WindowsAzure.Plugins.Connect.DomainOU	String
Microsoft.WindowsAzure.Plugins.Connect.DNSServers	String
Microsoft.WindowsAzure.Plugins.Connect.Administrators	String
Microsoft.WindowsAzure.Plugins.Connect.DomainSiteName	String

Figure 8-16. *Virtual network*

The settings configuration elements allow you to not only configure point-to-point connectivity, but also join the Windows Azure role instances to your on-premises domain.

Visual Studio Project Actions

Once you have created a Windows Azure cloud service project, you can work with the cloud service roles, work with storage services, or work on the debug and deployment of the cloud service.

Working with Cloud Service Roles

You can associate an existing Web role or a Worker role from a solution to the cloud service project, or create a new role by right-clicking on the `Roles` subdirectory and selecting Add, as shown in Figure 8-17.

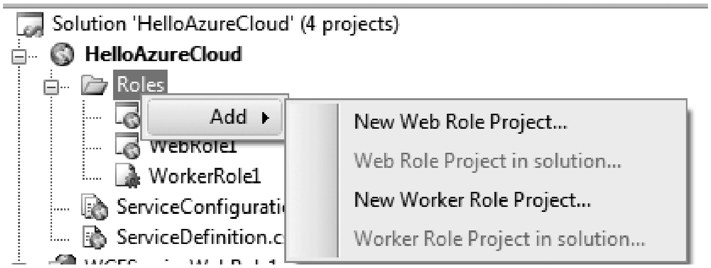
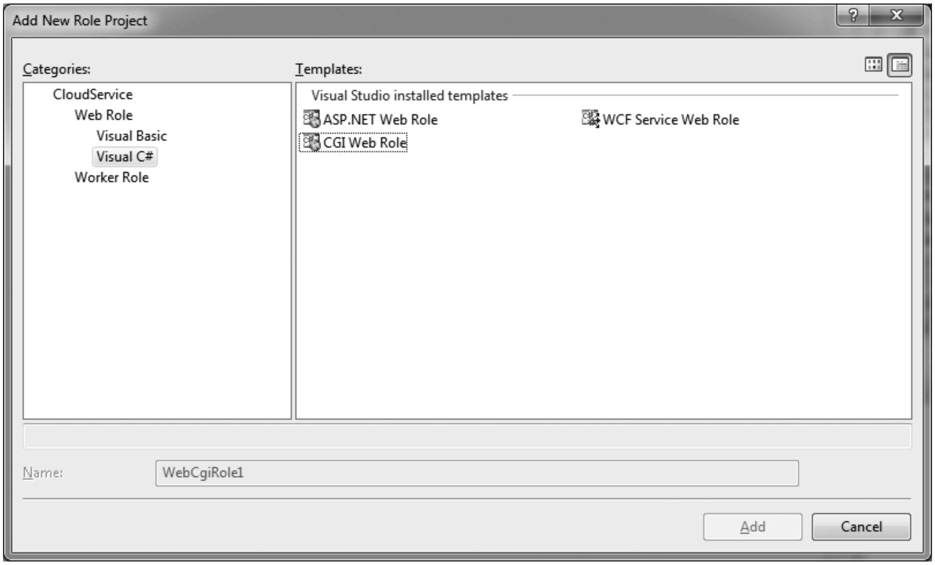

Solution 'HelloAzureCloud' (4 projects)
HelloAzureCloud
Roles
Add
WebRole1
WorkerRole1
ServiceConfigurati
ServiceDefinition.c
New Web Role Project...
Web Role Project in solution...
New Worker Role Project...
Worker Role Project in solution...

Figure 8-17. Adding associate roles to cloud service

By selecting New Web Role or New Worker Role project, you can create a new Web role project in the solution that is associated with the cloud service project. By selecting a Web role or Worker role project in the solution, you can associate an existing project in the solution to the cloud service project. Figure 8-18 shows option for adding a new role to the existing cloud service.



Name: WebCgiRole1

Figure 8-18. Adding new roles

Working with Storage Services

The Windows Azure development fabric includes a local storage environment that resembles the cloud storage service. It has development-specific blob, queue, and table services that simulate the ones in the Windows Azure cloud. These services depend on SQL Server 2005 or 2008 database. So, you need to have SQL Server 2005 or 2008 installed on your machine to work with the Storage services development environment (also called the development storage).

To start the development storage, select Start All Programs Windows Azure SDK Storage Emulator, as shown in Figure 8-19.

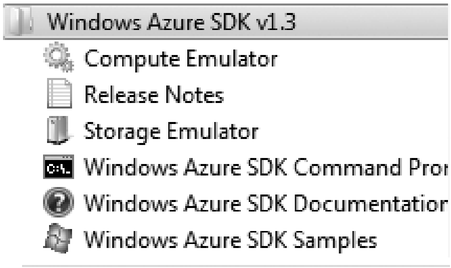

Windows Azure SDK v1.3
Compute Emulator
Release Notes
Storage Emulator
Windows Azure SDK Command Pror
Windows Azure SDK Documentatior
Windows Azure SDK Samples

Figure 8-19. Starting development storage

When you debug your service within Visual Studio, it starts the development storage, which you can access by right-clicking the Windows Azure system tray icon and selecting Show Storage Emulator UI. Figures 8-20 and 8-21 illustrate the system tray options and the development storage user interface. Figure 8-21 shows the local machine endpoints for each storage service.

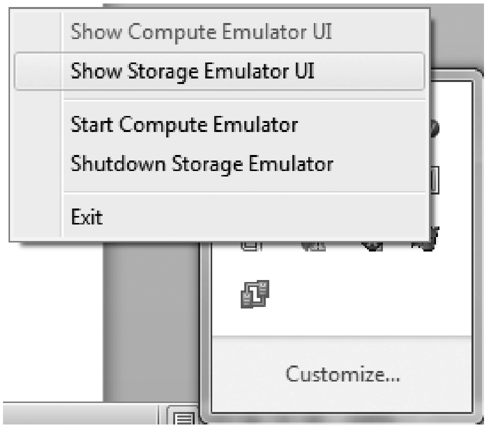
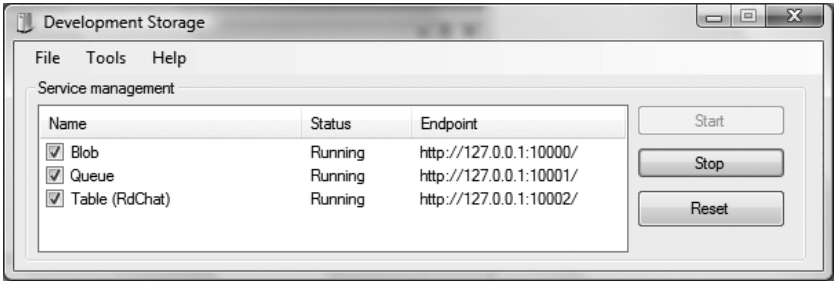


Figure 8-20. Windows Azure system tray options



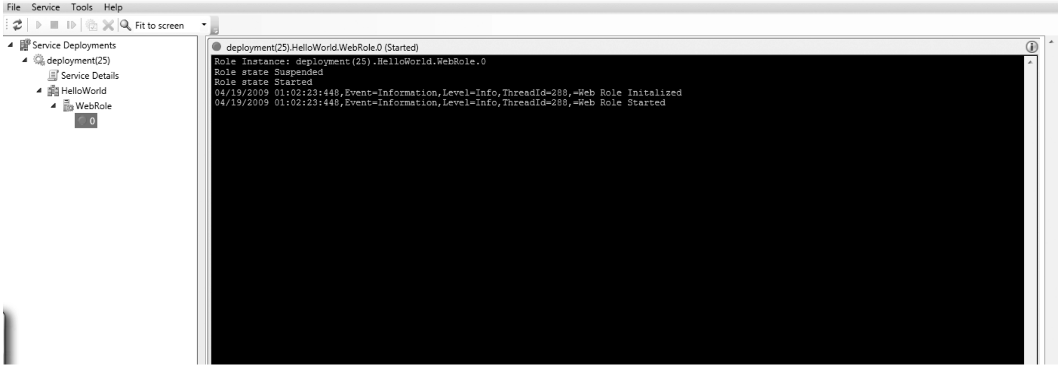
Name	Status	Endpoint
☑ Blob	Running	http://127.0.0.1:10000/
☑ Queue	Running	http://127.0.0.1:10001/
☑ Table (RdChat)	Running	http://127.0.0.1:10002/

Figure 8-21. Development storage user interface

Debugging in Visual Studio

In the Windows Azure Cloud environment, no direct debugging is available. You have two options: IntelliTrace and logging. IntelliTrace is available only in Visual Studio Ultimate, but it's worth the investment when developing for Windows Azure. IntelliTrace records a series of events and allows you to playback that recording with the existing code on your machine. This suits perfectly in cloud environments, because you cannot directly debug services running Windows Azure from Visual Studio. I have covered IntelliTrace later in the chapter.

In the development fabric, you can debug by adding breakpoints in the code and by viewing the logging information in the development fabric user interface. As with any .NET application, Visual Studio attaches the debugger to the application when run in debug mode in the development fabric. The debugger will break to the breakpoint set in the Web and Worker role projects. In the Windows Azure cloud, the Visual Studio debugging environment is not available, so one of the best options is to log. Logs can be automatically transferred to Windows Azure Storage and then collected via tools. I will discuss diagnostics and logging later in this chapter. Figure 8-22 illustrates the Development Fabric UI used for logging.

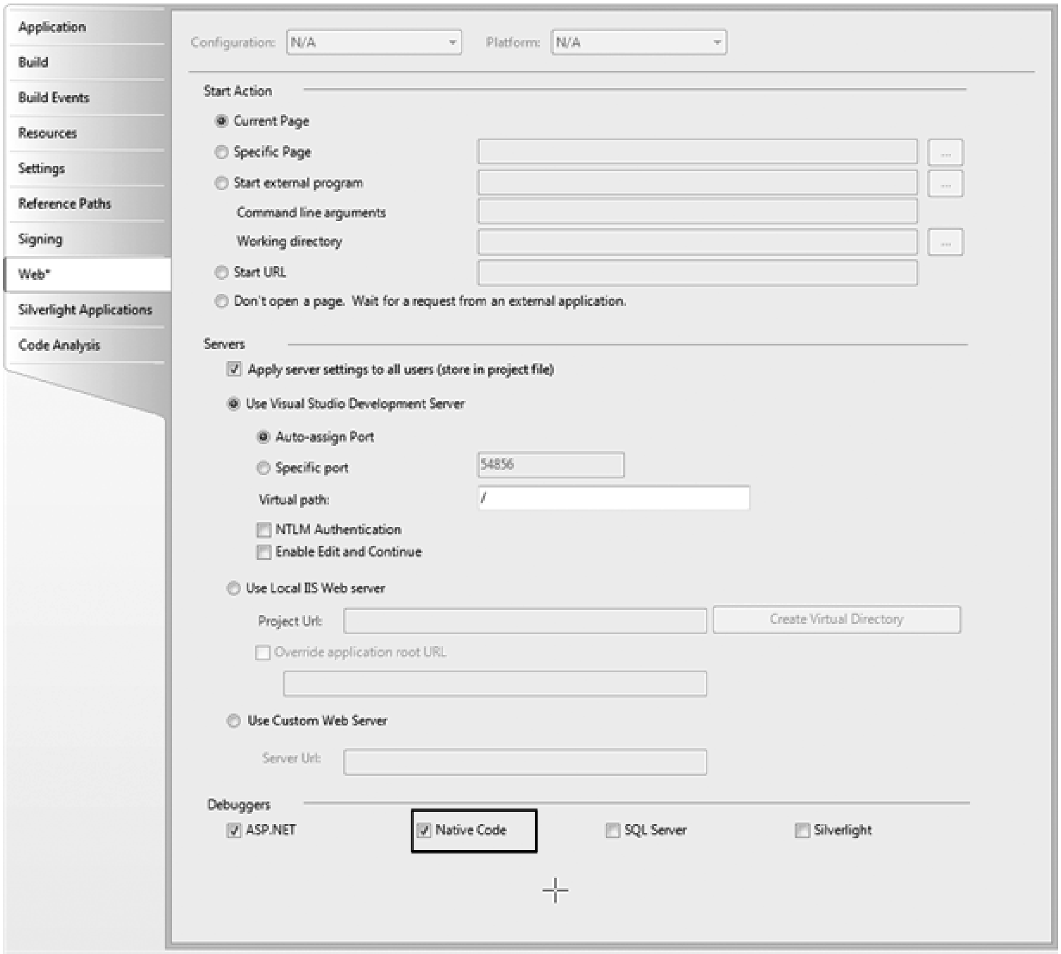


```
Role Instance: deployment(25).HelloWorld.WebRole.0
Role state Suspended
Role state Started
04/19/2009 01:02:23:448,Event=Information,Level=Info,ThreadId=288,=Web Role Initalized
04/19/2009 01:02:23:448,Event=Information,Level=Info,ThreadId=288,=Web Role Started
```

Figure 8-22. Development Fabric UI used for logging

■ **Tip** I recommend inserting logging statements to the Windows Azure application right from the beginning. This way, you can debug the application in the development fabric as well as in the Windows Azure cloud without making any code changes.

To enable native code debugging in a Web role project, right-click the Web role project, select Properties, go to the Web tab, and select the Native Code checkbox in the Debuggers section, as shown in Figure 8-23.



Configuration: N/A Platform: N/A

Start Action

● Current Page
○ Specific Page
○ Start external program
Command line arguments
Working directory
○ Start URL
○ Don't open a page. Wait for a request from an external application.

Servers

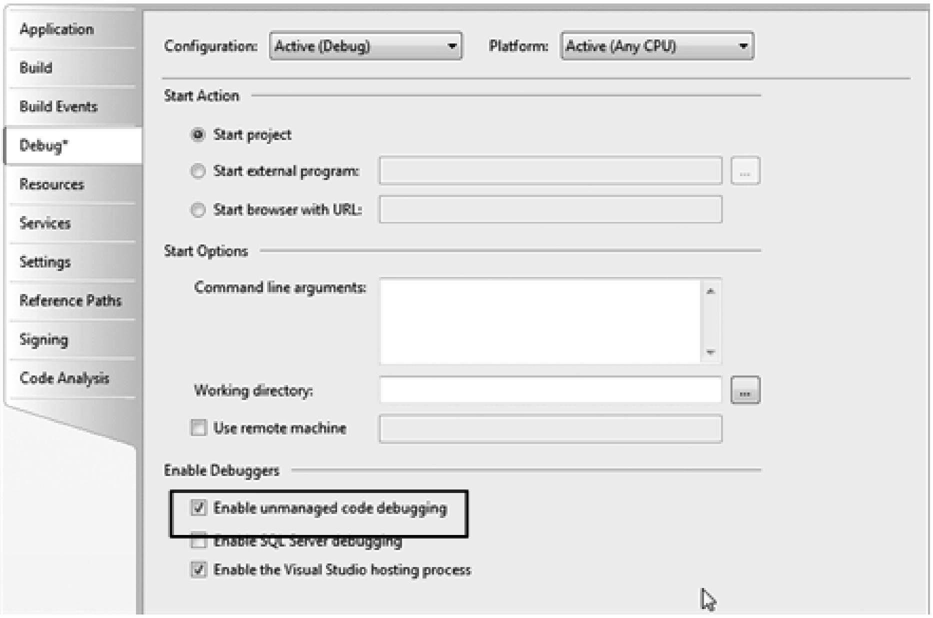
☑ Apply server settings to all users (store in project file)
● Use Visual Studio Development Server
● Auto-assign Port
○ Specific port 54856
Virtual path: /
□ NTLM Authentication
□ Enable Edit and Continue
○ Use Local IIS Web server
Project Url: Create Virtual Directory
□ Override application root URL
○ Use Custom Web Server
Server Url:

Debuggers

☑ ASP.NET ☑ Native Code □ SQL Server □ Silverlight

Figure 8-23. *Web role unmanaged code debugging*

To enable native code debugging in a Worker role project, right-click the Worker role project, select Properties, go to the Debug tab, and select the “Enable unmanaged code debugging” checkbox, as shown in Figure 8-24.



Configuration: Active (Debug)

Platform: Active (Any CPU)

Start Action

● Start project

○ Start external program:

○ Start browser with URL:

Start Options

Command line arguments:

Working directory:

□ Use remote machine

Enable Debuggers

☑ Enable unmanaged code debugging

□ Enable SQL Server debugging

☑ Enable the Visual Studio hosting process

Figure 8-24. Worker role unmanaged code debugging

Packaging the Service

To deploy the Windows Azure cloud service in the cloud, you have to package it into a `.cspkg` file containing all the assemblies and components, and upload the package to Windows Azure developer portal. To package a service, right-click the cloud service project, and select Publish, as shown in Figure 8-25.

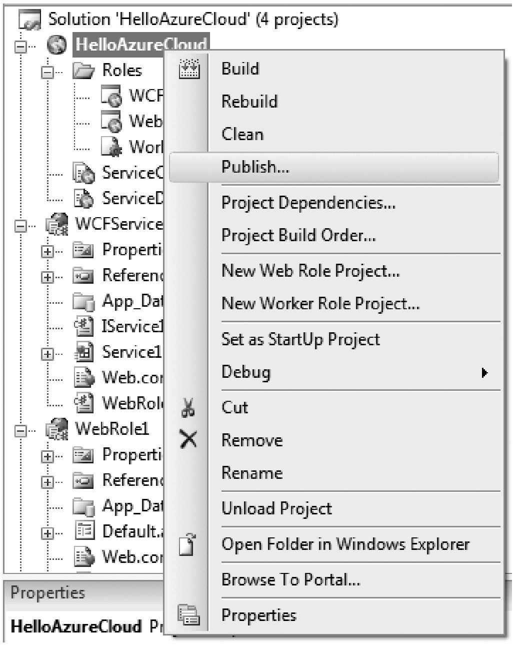
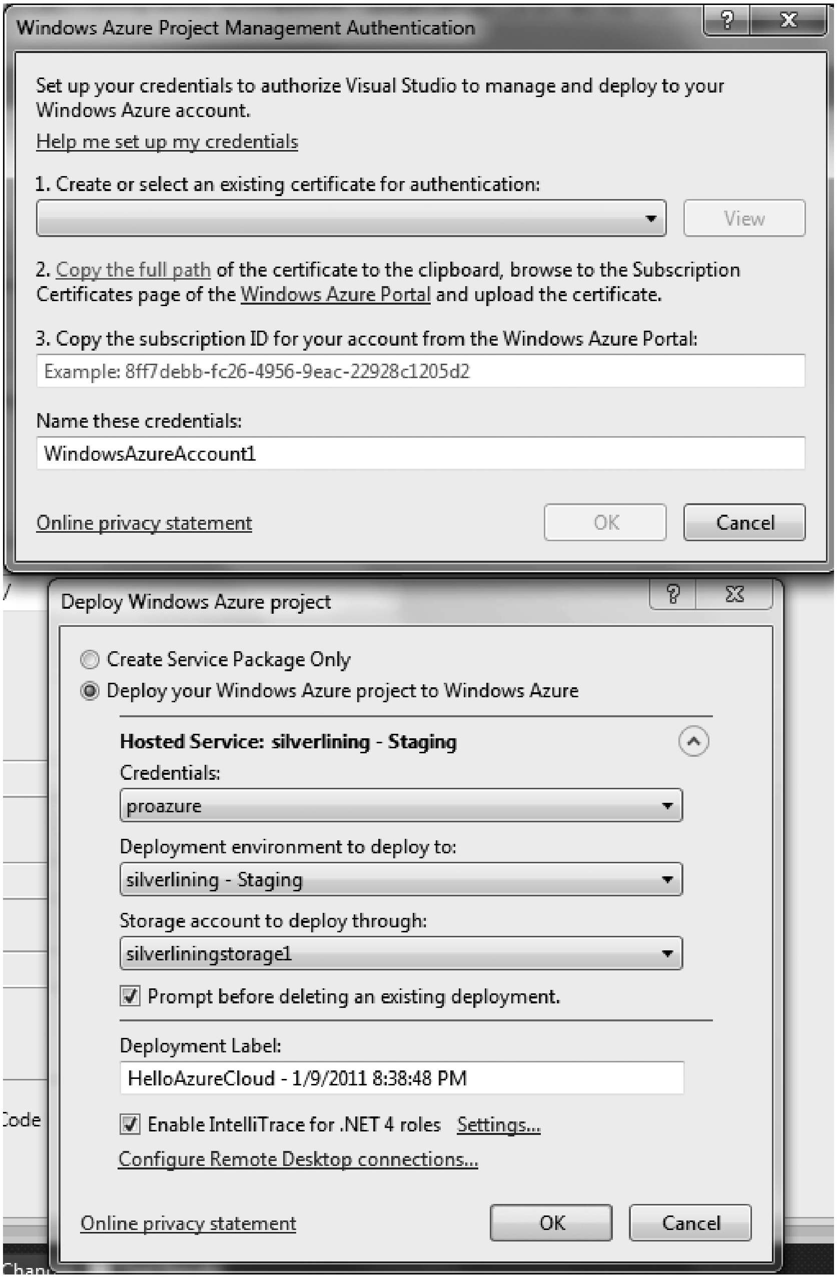


Figure 8-25. *Packaging a Windows Azure service*

When you select Publish, Visual Studio opens up Deploy Windows Azure project dialog box, as shown in Figure 8-26.



Windows Azure Project Management Authentication

Set up your credentials to authorize Visual Studio to manage and deploy to your Windows Azure account.

Help me set up my credentials

1. Create or select an existing certificate for authentication:

2. Copy the full path of the certificate to the clipboard, browse to the Subscription Certificates page of the Windows Azure Portal and upload the certificate.

3. Copy the subscription ID for your account from the Windows Azure Portal:

Example: 8ff7debb-fc26-4956-9eac-22928c1205d2

Name these credentials:

WindowsAzureAccount1

Online privacy statement

Deploy Windows Azure project

○ Create Service Package Only

● Deploy your Windows Azure project to Windows Azure

Hosted Service: silverlining - Staging

Credentials: proazure

Deployment environment to deploy to: silverlining - Staging

Storage account to deploy through: silverliningstorage1

☑ Prompt before deleting an existing deployment.

Deployment Label: HelloAzureCloud - 1/9/2011 8:38:48 PM

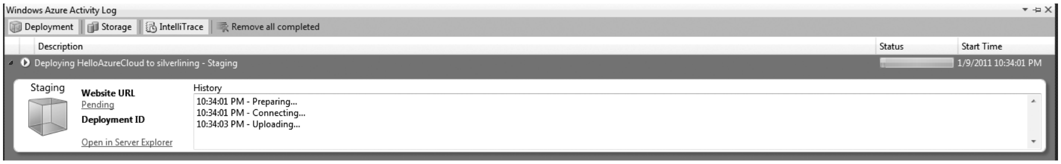
☑ Enable IntelliTrace for .NET 4 roles Settings...

Configure Remote Desktop connections...

Online privacy statement

Figure 8-26. *Deploy Windows Azure*

Visual Studio lets you deploy the project directly from Visual Studio. Though it is not recommended to deploy production cloud services from a developer's desktop, this is a handy feature for deploying cloud services in testing and staging environments. To deploy cloud services from Visual Studio, you need to create an X.509 certificate and upload it to your account in the Management portal. Visual Studio uses the Windows Azure Service Management API for deploying cloud services, and an X.509 certificate is required for accessing the Service Management API. Visual Studio lets you create a new certificate or choose an already existing one from the Credentials dropdown. Once you create or select a certificate, you can then provide a subscription ID of your account, which is needed by the Service Management API. Once you specify the credentials, you can choose your hosted service from the dropdown with the deployment slot (staging or production), the storage account for uploading the deployment package and a deployment label. You can also select if you want to enable IntelliTrace. When you click OK, Visual Studio opens up a Windows Azure Activity Log window, as shown in Figure 8-27, and shows displays the status of your deployment.



Deploying HelloAzureCloud to silverlining - Staging 1/9/2011 10:34:01 PM

Staging

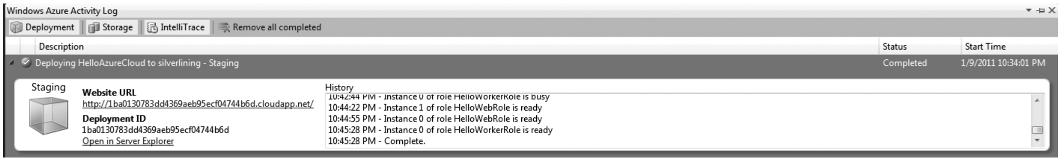
Website URL
Pending
Deployment ID
Open in Server Explorer

History

```
10:34:01 PM - Preparing...
10:34:01 PM - Connecting...
10:34:03 PM - Uploading...
```

Figure 8-27. Windows Azure Activity Log

Once the deployment is complete, the activity log will display the URL of the service, as shown in Figure 8-28.



Deploying HelloAzureCloud to silverlining - Staging Completed 1/9/2011 10:34:01 PM

Staging

Website URL
http://1ba0130783dd4369aeb95ecf04744b6d.cloudapp.net/
Deployment ID
1ba0130783dd4369aeb95ecf04744b6d
Open in Server Explorer

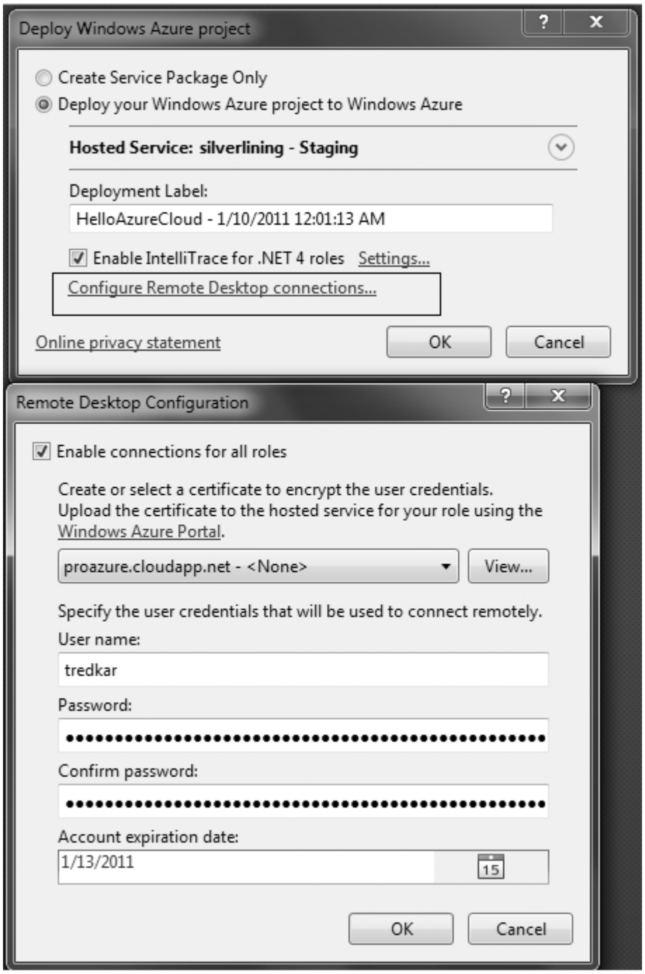
History

```
10:42:44 PM - Instance 0 of role HelloWorkerRole is busy
10:44:22 PM - Instance 1 of role HelloWebRole is ready
10:44:55 PM - Instance 0 of role HelloWebRole is ready
10:45:28 PM - Instance 0 of role HelloWorkerRole is ready
10:45:28 PM - Complete.
```

Figure 8-28. Windows Azure Activity Log on completion

On the Deploy Windows Azure Project dialog box, you also have the option to only create the service package. Visual Studio creates two files: `[Service Name].cspkg` and `ServiceConfiguration.cscfg`. The `[Service Name].cspkg` is the service package containing all the service components required by Windows Azure to run the service in the cloud. The `.cspkg` file is a zip archive, and you can explore its contents by renaming it to `.zip` and extracting it. The `ServiceConfiguration.cscfg` file is the configuration file for the service instances. It is a copy of the `ServiceConfiguration.cscfg` file from the cloud service project. You can deploy the `.cspkg` either by manually uploading to the Management portal or through PowerShell scripts that use the Service Management API. In a typical software development life cycle, the release management team will deploy the package using custom scripts that call the Service Management API.

You can also configure Remote Desktop Connections for your roles from the Deploy Windows Azure Project dialog box. As shown in Figure 8-29, the remote desktop configuration requires an X.509 certificate, username, and password for remote login, and an expiration date for the login.



Deploy Windows Azure project

○ Create Service Package Only

● Deploy your Windows Azure project to Windows Azure

Hosted Service: silverlining - Staging

Deployment Label:
HelloAzureCloud - 1/10/2011 12:01:13 AM

☑ Enable IntelliTrace for .NET 4 roles Settings...

Configure Remote Desktop connections...

Online privacy statement

OK Cancel

Remote Desktop Configuration

☑ Enable connections for all roles

Create or select a certificate to encrypt the user credentials.
Upload the certificate to the hosted service for your role using the Windows Azure Portal.

proazure.cloudapp.net - <None> View...

Specify the user credentials that will be used to connect remotely.

User name:
tredkar

Password:
●●

Confirm password:
●●

Account expiration date:
1/13/2011

OK Cancel

Figure 8-29. *Remote desktop configuration*

■ **Note** The password accepted by the Remote Desktop Connection needs to be encrypted. The procedure for encrypting the passwords is listed here `http://msdn.microsoft.com/en-us/library/gg432965.aspx`.

Windows Azure SDK Tools

The Windows Azure SDK tools are located in the directory `C:\Program Files\Windows Azure SDK\v1.0\bin` for a default Windows Azure installation. Table 8-2 lists the tools included in the Windows Azure SDK.

Table 8-2. *Windows Azure SDK Tools*

Tool	Description
`CSPack.exe`	This tool is used to package a service for deployment. It takes in a `ServiceDefinition.csdef` file and outputs a `.cspkg` file.
`CSRun.exe`	This tool deploys a service into the local development fabric. You can also control the run state of the development fabric from this tool. This tool depends on the service directory structure created by the `CSPack.exe /copyonly` option.
`Csupload.exe`	`CSUpload.exe` is used for uploading VHD images to the Windows Azure Management Portal.
`DSInit.exe`	This tool initializes the development storage environment. It is automatically called by Visual Studio and DevelopmentStorage.exe when you run a cloud application in the development fabric for the first time.

Service Models

A service model of Windows Azure cloud service consists of two main configuration files: `ServiceDefinition.csdef` and `ServiceConfiguration.cscfg`. `ServiceDefinition.csdef` defines the metadata and configuration settings for the service. `ServiceConfiguration.cscfg` sets the values of configuration settings for the runtime instance of the service. You can modify the `ServiceConfiguration.cscfg` while the service is running, but modification to `ServiceDefinition.csdef` requires an update to the service package. The overall service model defines the metadata and configuration parameters and the end state of the service. Windows Azure reads these files when deploying instances of your service in the cloud and the Fabric Controller takes appropriate deployment actions.

ServiceDefinition.csdef

The `ServiceDefinition.csdef` file defines the overall structure of the service. It defines the roles available to the service, the input/external endpoints, web sites, local storage, plug-in modules, and certificates. With Windows Azure SDK version 1.3, Microsoft introduced a plug-in model for Windows Azure features. For example, diagnostics, remote desktop, and Windows Azure Connect are plug-in modules specified in the `<Imports>` elements in Listing 8-1. It also defines the custom configuration settings for the service. The values of these configuration parameters are set in the `ServiceConfiguration.cscfg` file. Listing 8-1 shows the contents of a `ServiceDefinition.csdef` file.

Listing 8-1. *ServiceDefinition.csdef*

```
<?xml version="1.0" encoding="utf-8"?>
<ServiceDefinition name="HelloAzureCloud"
xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceDefinition">
  <WebRole name="HelloWebRole" enableNativeCodeExecution="true">
    <Sites>
      <Site name="Web">
        <Bindings>
          <Binding name="HttpIn" endpointName="HttpIn" />
        </Bindings>
      </Site>
    </Sites>
    <LocalResources>
      <LocalStorage name="HelloAzureWorldLocalCache" sizeInMB="10" />
    </LocalResources>
    <ConfigurationSettings>
      <Setting name="PerformanceCounters" />
      <!--This is the current logging level of the service -->
      <Setting name="LogLevel" />
      <Setting name="ThrowExceptions" />
      <Setting name="EnableOnScreenLogging" />
    </ConfigurationSettings>
    <Endpoints>
      <InputEndpoint name="HttpIn" protocol="http" port="8080" />
    </Endpoints>
    <Imports>
      <Import moduleName="Diagnostics" />
      <Import moduleName="RemoteAccess" />
    </Imports>
    <Certificates>
    </Certificates>
  </WebRole>
  <WorkerRole name="HelloWorkerRole" enableNativeCodeExecution="true">
    <Endpoints>
      <!-- Defines an internal endpoint for inter-role communication that can be used to
communicate between worker or Web role instances -->
      <InternalEndpoint name="MyInternalEndpoint" protocol="tcp" />
      <!-- This is an external endpoint that allows a role to listen on external
communication, this could be TCP, HTTP or HTTPS -->
      <InputEndpoint name="MyExternalEndpoint" port="9001" protocol="tcp" />
    </Endpoints>

    <Imports>

      <Import moduleName="Diagnostics" />

      <Import moduleName="RemoteAccess" />

      <Import moduleName="RemoteForwarder" />

      <Import moduleName="Connect" />

    </Imports>
    <ConfigurationSettings>
      <Setting name="PerformanceCounters" />
    </ConfigurationSettings>
    <Startup>
      <Task commandLine="Startme.cmd" executionContext="limited" taskType="simple">
      </Task>
    </Startup>
  </WorkerRole>
</ServiceDefinition>
```

Listing 8-1 is the service definition for the HelloAzureCloud. It has a Web role and a Worker role instance defined. The <LocalStorage> element defines the local storage space for the service role. The <ConfigurationSettings> element defines some custom settings for the service role.

■ **Note** The ServiceDefinition.csdef file of a service cannot be changed at runtime, because it defines the shape and non-changeable parameters of the service. You have to republish the service after changing its ServiceDefinition.csdef for the changes to take effect. For more details on the ServiceDefinition.csdef schema, please visit http://msdn.microsoft.com/en-us/library/dd179395.aspx.

Endpoints

Windows Azure roles can have two types of endpoints: internal and input. The internal endpoints are used for inter-role communications within the same cloud service, whereas the input endpoints can be accessed from anywhere. Figure 8-30 illustrates an example of internal endpoints.

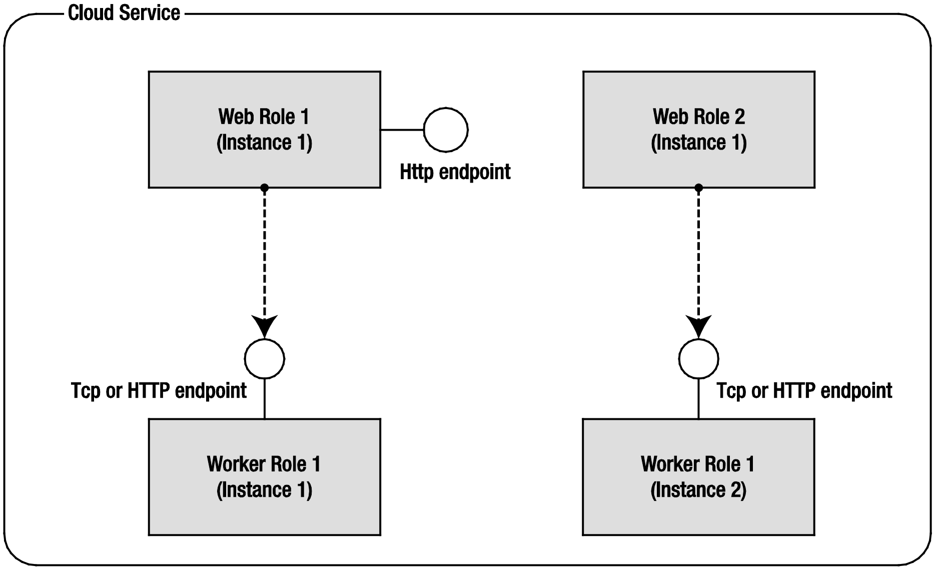


Figure 8-30. *Internal endpoints for inter-role communication*

In Figure 8-30, there are two Web roles with one instance each and two instances of a Worker role. The Worker role exposes an endpoint that is consumed by both the Web roles. Note that each Web role can communicate with the exact instance of the Worker role. The Web role also exposes an HTTP endpoint that can be consumed by any of the roles in the cloud service. The endpoint only publishes the IP address and port of the instance; you still have to write TCP or HTTP service code for sending and receiving requests. You can get a reference to the internal endpoint of an instance as follows:

```
IPEndPoint internale = RoleEnvironment.Roles["HelloWorkerRole"].Instances[0]
.InstanceEndpoints["MyInternalEndpoint"].IPEndpoint;
```

where `HelloWorkerRole` is the name of the Worker role and `MyInternalEndpoint` is the name of the endpoint. You can get the IP address of an internal instance end point in the following manner:

```
string ipaddress =  RoleEnvironment.Roles["HelloWorkerRole"].Instances[0]
.InstanceEndpoints["MyInternalEndpoint"].IPEndpoint.ToString();
```

Figure 8-31 illustrates the input endpoints of a Web role and a Worker role.

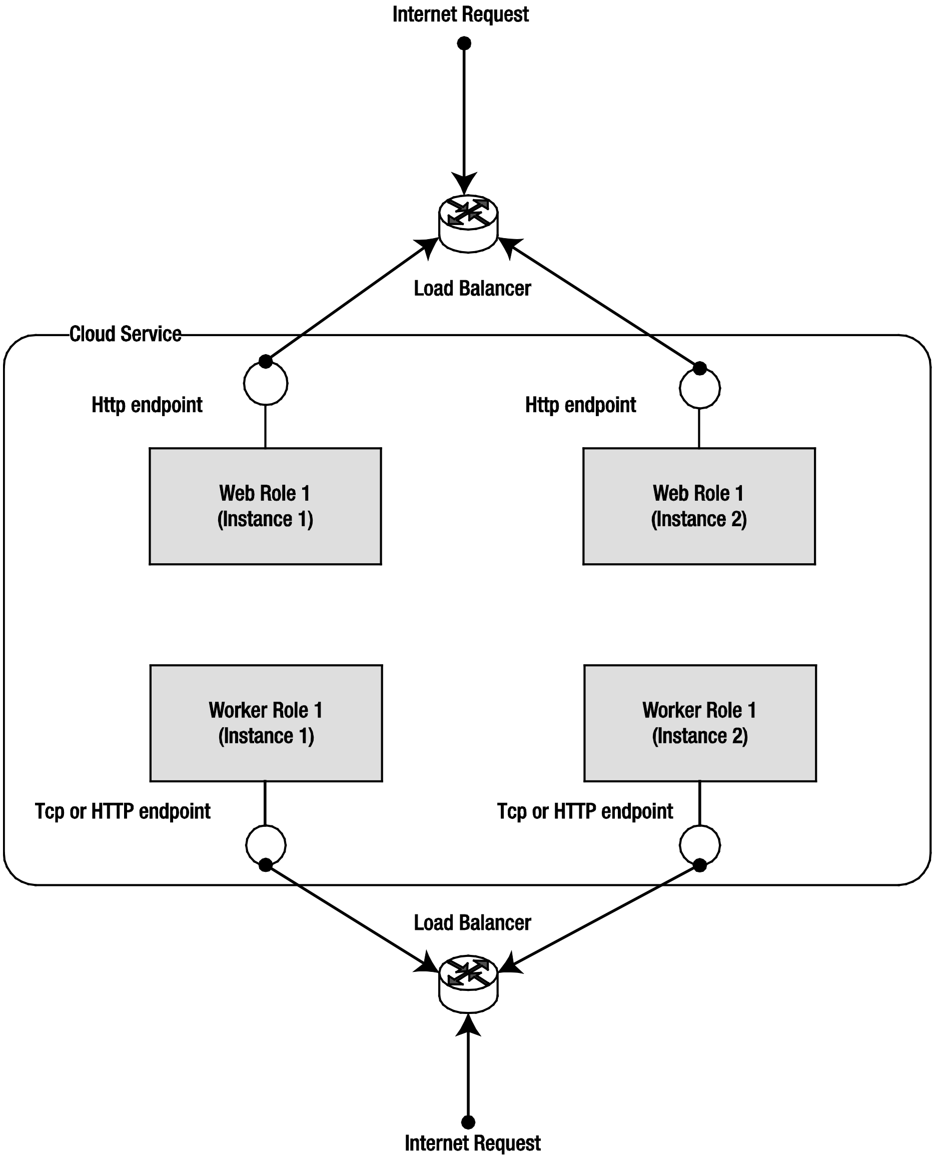


Figure 8-31. *Input endpoints for external communication*

The Web role instances have default HTTP input endpoints for accepting Internet requests. Windows Azure also allows Worker roles to have HTTP and TCP input endpoints for accepting connections over the Internet. Like the internal endpoint, the access to input endpoint is not limited within the cloud service; any external application can communicate with the input endpoint of the role. In Figure 8-31, Web Role 1 and Worker Role 1 have input endpoints available for communication. Any application can now communicate with endpoints of these roles over the Internet. Because the input endpoints are exposed externally, they are automatically load-balanced by Windows Azure between instances. In some documentation, input endpoints are also referred to as external endpoints. You can get a reference to the input endpoint of an instance as follows:

```
IPEndPoint inpute = RoleEnvironment.Roles["HelloWorkerRole"].Instances[0]

.InstanceEndpoints["MyInputEndpoint"].IPEndpoint;
```

where `HelloWorkerRole` is the name of the Worker role and `MyInputEndpoint` is the name of the endpoint. Once you have the `IPEndPoint` object, you can get the IP address and port number of the endpoint to initiate communications.

Note You can run multiple web sites on one Web Role instance. For more information, below are two blogs I recommend:

By Wade Wegner, www.wadewegner.com/2011/02/running-multiple-websites-in-a-windows-azure-web-role/

By Andy Cross, blog.bareweb.eu/2011/01/azure-running-multiple-web-sites-in-a-single-webrole/

Local Storage

Windows Azure runtime provides a static function `LocalResource GetLocalResource (string localResourceName)` in the `Microsoft.WindowsAzure..ServiceRuntime.RoleEnvironment` class to get reference to the `LocalResource` class, which represents the local storage space reserved for the service. The `localResourceName` function parameter is the name of storage space defined as the name attribute of `<LocalStorage>` element. In Listing 3-1, I am allocating a space of 10MB for the storage space named `L1` on local machine of the service role instance. I can now get a reference to the local storage space by calling the function `LocalResource resource = RoleEnvironment.GetLocalResource("L1");` and calls System.IO file operations on the local path.

■ **Caution** Local storage space allocated on the local machine is local for that instance. If the `cleanOnRoleRecycle` attribute is set to `false`, the data from local directory will be lost on role recycle. So, while developing applications, you should consider local storage purely for unreliable caching purposes with data loss checks built into the application.

Startup Tasks

The `<Startup>` element defines startup tasks that can be executed while the role is starting. Typical scenarios for using startup tasks are as follows:

- Pre-installation of custom software before the role starts (e.g., anti-virus software, Java virtual machine, and so on)
- Installation of COM DLLs
- Making changes to the IIS or any Windows Service before the cloud service is deployed

■ **Note** It is important to understand that role configurations do not survive upgrades and reboots. Therefore, startup tasks will be executed during every startup cycle for maintaining the consistency of instance configuration.

Listing 8-2 illustrates the format for using the startup tasks.

Listing 8-2. *ServiceDefinition.csdef*

```
<WebRole name=" HelloWebRole ">
  <Startup>
    <Task commandline="[relative path of the executable file]"
          executionContext="limited|elevated"
          taskType="simple|foreground|background"/>
  </Startup>
</WebRole>
```

The `executionContext` attribute defines the permissions under which the command will be run. The value “limited” indicates the task will be run under the same permission sets as the role, i.e., in user context. The value “elevated” indicates the task will be run under administrator’s privileges. Elevated tasks are typically used for modifying system-level resources like registry and Windows Services. The `taskType` attribute specifies the behavior of the role while the task is executing. Simple task type is the default value and blocks the role till the task finishes its execution. This task type is commonly used in installing pre-requisite applications like anti-virus software or application frameworks. The background task type does not block the role instantiation while the task is in progress. The task is executed as a background task. The foreground task does not block the role instantiation but blocks the recycle of the role until the task finishes.

■ **Note** In Windows Azure SDK 1.5 (September 2011), new features were included for specifically for launching executables and including local folder contents in application packages.

Sample schema for adding these to your ServiceDefinition file is listed below.

```
<Runtime executionContext="[limited/elevated]"> <Environment> <Variable name="<variable-name>"
value="<variable-value>">  <RoleInstanceValue
```

```
xpath="<xpath-to-role-environment-settings>"/>  </Variable> </Environment> <EntryPoint>
<NetFxEntryPoint assemblyName="<name-of-assembly-containing-
entrypoint>"targetFrameworkVersion="<.net-framework-version>"/>  <ProgramEntryPoint
commandLine="<application>" setReadyOnProcessStart="[true/false] ""/> </ EntryPoint></Runtime>
```

You can find more information on these features here: msdn.microsoft.com/en-us/library/gg441573(MSDN.10).aspx

Nathan Totten also has a good blog article on this topic at ntotten.com/2011/09/running-processes-in-windows-azure/

■ **Tip** You can also run an entire role with elevated privileges by adding the Runtime element with elevated execution context in your role definition. This is not recommended, but there might be some background applications that need elevated privileges. Make sure you don't have any input endpoints defined on such role instances.

```
<WebRole name="WebRole1">
    <Runtime executionContext="elevated" />
  </WebRole>
```

Full IIS Support

From Windows Azure 1.3 SDK, you have access to Full IIS running on Windows Azure Web roles. You can run multiple web sites pointing to the same physical directory but separated by host headers. For example, in Listing 8-3, there are two sites names, A1 and A2, that are bound to the same HttpIn endpoint, but separated by different host headers. You can also create a virtual applications and virtual directories under the site for further separating the applications within the site.

Listing 8-3. *Full IIS Virtual Directories*

```
<WebRole name="MyWebApp">
  <Sites>
    <Site name="A1" physicalDirectory="..\MyWebApp">
      <Bindings>
        <Binding name="HttpIn"
                 endpointName="HttpIn"
```

```
                hostHeader="www.a1.com" />
      </Bindings>
    </Site>
    <Site name="Contoso" physicalDirectory="..\MyWebApp">

<VirtualApplication name="myapp"
 physicalDirectory="..\..\..\..\apps\myapp">

<VirtualDirectory name="Styles"
 physicalDirectory="..\MyWebApp\Styles" />
 </VirtualApplication>
      <Bindings>
        <Binding name="HttpIn"
                endpointName="HttpIn"
                hostHeader="www.a2.com" />
      </Bindings>
    </Site>
  </Sites>
  ...
</WebRole>
```

Full Trust Execution

By default, Windows Azure applications run under full trust in the cloud environment. When running under partial trust, the code has access only to limited resources and libraries. When running under full trust, cloud services can access certain system resources and can call managed assemblies as well as native code. To enable Full Trust in your application, set the `enableNativeCodeExecution` attribute of the `<WebRole>` or `<WorkerRole>` element in the `ServiceDefinition.csdef` file to `true`:

```
<WebRole name="<role name>" enableNativeCodeExecution="true|false">
```

Table 8-3 lists the permissions for a cloud application role running in partial and full trust execution modes.

Table 8-3. *Partial and Full Trust Permissions*

Resource	Partial Trust	Full Trust
Call managed code assemblies	Assemblies with `AllowPartiallyTrustedCallers` attribute	All assemblies

System registry	No access	Read access to
		`HKEY_CLASSES_ROOT`
		`HKEY_LOCAL_MACHINE`
		`HKEY_USERS`
		`HKEY_CURRENT_CONFIG`

32-bit P/Invoke	Not supported	Not supported
64-bit P/Invoke	Not supported	Supported
32-bit native sub-process	Not supported	Supported
64-bit native sub-process	Not supported	Supported
Local storage	Full access	Full access
System root and its subdirectories	No access	No access
Windows (e.g., `C:\Windows`) and its subdirectories	No access	Read access
Machine configuration files	No access	No access
Service configuration file (`ServiceConfiguration.cscfg`)	Read access	Read access

■ **Note** You can find more information on the Windows Azure partial trust policy in the Windows Azure SDK documentation at `http://msdn.microsoft.com/en-us/library/dd573355.aspx`.

Table 8-3 clearly shows that in partial trust you cannot call native code and the access to the machine resources are limited. Even in full trust execution, the access has been limited to prevent any system-related damage. Partial trust application roles can call only managed code assemblies that have `AllowPartiallyTrustedCallers` attribute, whereas a full trust application role can all any managed code assembly. A partial trust application role cannot make any P/Invoke native calls. A full trust application role can make P/Invoke calls to a 64-bit library. P/Invoke calls to a 32-bit library are not directly supported in Windows Azure. Instead, you could spawn a 32-bit sub-process from your application role and make P/Invoke calls to 32-bit library from within that sub-process. The system root directory (usually `C:\Windows\system32`) is not accessible in Windows Azure. A full trust application role has only read access to the Windows directory (usually `C:\Windows`). Both, full and partial trust roles have full access to the local storage. Local storage is the recommended temporary file and data storage for Windows Azure applications.

■ **Caution** The resource access works differently in the Windows Azure cloud and the development fabric. In the Windows Azure cloud, the application role runs under the privileges of a standard Windows Azure account, whereas the application role in the development fabric runs under the logged-in user account. So, the application role running in the local development fabric may behave differently to the same application role running in the Windows Azure cloud environment.

Certificate Management

In Windows Azure, you can use certificates not only for encrypting the HTTPS endpoints of your Web and Worker roles but also for custom message level encryption. You can upload X.509 certificated to your Windows Azure service either from the Windows Azure Management Portal or using the Service Management API. You can upload any number of certificates for the service and these certificates will be installed in the Windows certificate stores of the role instances.

Once a certificate is uploaded to the service, it can be referenced in the `ServiceDefinition.csdef` and `ServiceConfiguration.cscfg`. The `ServiceDefinition.csdef` defines the name, store location, and store name of the certificate on the instance as shown here:

```
<Certificate name="C1" storeLocation="LocalMachine" storeName="My" />
```

The `ServiceConfiguration.cscfg` file defines the thumbprint and the thumbprint algorithm of the certificate as shown here.

```
<Certificate name="Certificate1" thumbprint="5CA27AF00E1759396Cxxxxxxxxxxxxx"
thumbprintAlgorithm="sha1" />
```

ServiceConfiguration.cscfg

The `ServiceConfiguration.cscfg` file contains the values for the configuration parameters that apply to one or more instance of the service. It also consists of the <Instances> element for scaling your service up and down. For each `<Import moduleName="">`, a series of configuration elements are automatically created in the configuration file for configuring the specific plug-in. You have to configure each plug-in by setting the values of these configuration elements. The service configuration file can be changed dynamically either from the Management portal or calling the Service Management API. Listing 8-4 shows the contents of the `ServiceConfiguration.cscfg` file corresponding to the `ServiceDefinition.csdef` file from Listing 8-1.

Listing 8-4. *ServiceConfiguration.cscfg*

```
<?xml version="1.0"?>

<ServiceConfiguration serviceName="HelloAzureCloud"
xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConfiguration">
  <Role name="HelloWebRole">

    <Instances count="2" />

    <ConfigurationSettings>
      <!--This is the current logging level of the service -->
      <!--Supported Values are Critical, Error,Warning,Information,Verbose-->
      <Setting name="LogLevel" value="Information" />
      <Setting name="ThrowExceptions" value="true" />
      <Setting name="EnableOnScreenLogging" value="true" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString"
value="UseDevelopmentStorage=true" />
      <Setting name="PerformanceCounters" value="\Processor(_Total)\% Processor
Time,\Memory\Available MBytes" />
      <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.Enabled" value="true" />
      <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountUsername"
value="tredkar" />
      <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountEncryptedPassword"
value="zBT3zHKb" />
      <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountExpiration"
value="2011-01-13T23:59:59.0000000-08:00" />
    </ConfigurationSettings>
    <Certificates>
      <Certificate name="Microsoft.WindowsAzure.Plugins.RemoteAccess.PasswordEncryption"
thumbprint="AD2D6E79DF99F3C5A55CD98FDBD7DB92F91BE4A7" thumbprintAlgorithm="sha1" />
    </Certificates>
  </Role>
  <Role name="HelloWorkerRole">

    <Instances count="1" />

    <ConfigurationSettings>
      <Setting name="Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString"
value="UseDevelopmentStorage=true" />
      <Setting name="PerformanceCounters" value="\Processor(_Total)\% Processor
Time,\Memory\Available MBytes" />
      <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.Enabled" value="true" />
      <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountUsername"
value="tredkar" />
      <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountEncryptedPassword"
value="MIIBH" />
      <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountExpiration"
value="2011-01-13T23:59:59.0000000-08:00" />
      <Setting name="Microsoft.WindowsAzure.Plugins.RemoteForwarder.Enabled" value="true" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.ActivationToken" value="0042eeb8-
e1a4-40e3-9141-3dbc7a24f135" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.Refresh" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.Diagnostics" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.WaitForConnectivity" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.EnableDomainJoin" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.DomainFQDN" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.DomainControllerFQDN" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.DomainAccountName" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.DomainPassword" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.DomainOU" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.DNSServers" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.Administrators" value="" />
      <Setting name="Microsoft.WindowsAzure.Plugins.Connect.DomainSiteName" value="" />
    </ConfigurationSettings>
    <Certificates>
      <Certificate name="Microsoft.WindowsAzure.Plugins.RemoteAccess.PasswordEncryption"
thumbprint="AD2D6E79DF99F3C5A55CD98FDBD7DB92F91BE4A7" thumbprintAlgorithm="sha1" />
    </Certificates>
  </Role>
</ServiceConfiguration>
```

In Listing 3-2, there are three roles defined, two Web roles and a Worker role. Each role one has only one instance.

■ **Note** For more details on the `ServiceConfiguration.cscfg` and ServiceDefinition.csdef schemas, please visit http://msdn.microsoft.com/en-us/library/dd179398.aspx.

Web.config Versus ServiceConfiguration.cscfg

The `web.config` file is the configuration file for an ASP.NET web application. It defines the behavior of the ASP.NET application and also custom configuration setting values. Configuration setting values in `web.config`, or `app.config`, for that matter, cannot be dynamically changed at runtime and made available to the application without redeploying the application. Similarly, the ASP.NET behavior defined in a `web.config` file cannot be included in a `ServiceConfiguration.cscfg` file.

`ServiceConfiguration.cscfg` is meant purely for storing configuration-setting values used by the service at runtime; it is not meant to define the behavior of the ASP.NET runtime or the .NET Runtime. `ServiceConfiguration.cscfg` is the configuration file for one or more instances of a Windows Azure service deployed in the cloud. You can change the configuration values in `ServiceConfiguration.cscfg` at runtime, and they will be available to the application without redeploying the application to the cloud.

Development Fabric

The development fabric simulates the Windows Azure cloud runtime environment on your local machine. The development fabric is specifically designed for development and testing in your local environment. You cannot attach a development fabric with the Windows Azure cloud service. The development fabric user interface can be started in any of the following manners:

- By debugging or running a cloud service from within Visual Studio
- By running `CSRun.exe` from the command line with valid parameters
- By running `DFUI.exe` from the Windows Azure SDK `bin` directory
- From the Windows Azure SDK programs Start menu

Once the development fabric starts, you can access it from the development fabric system tray icon. Figure 8-32 illustrates the development fabric user interface hosting a cloud service.

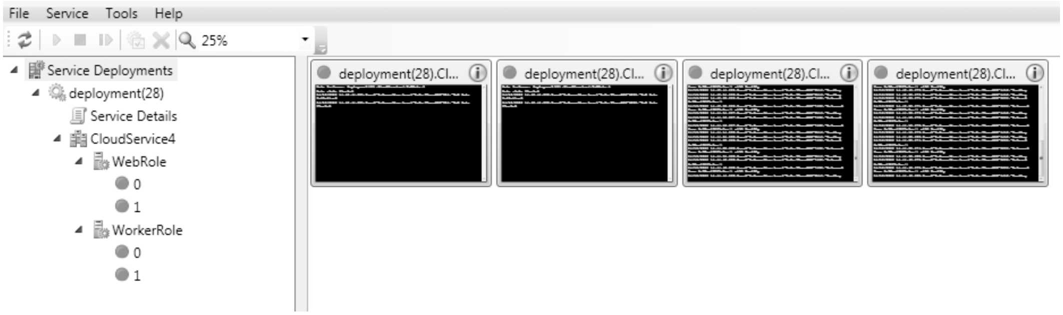


Figure 8-32. Development fabric UI

The development fabric UI shows the service deployments in the local environment and allows you to alter the state of a running service. You can run, suspend, restart, or remove a service deployment from within the development fabric UI.

In the development fabric, you can attach a debugger to the running instance at runtime by right-clicking one of the instance and selecting Attach Debugger, as shown in Figure 8-33.

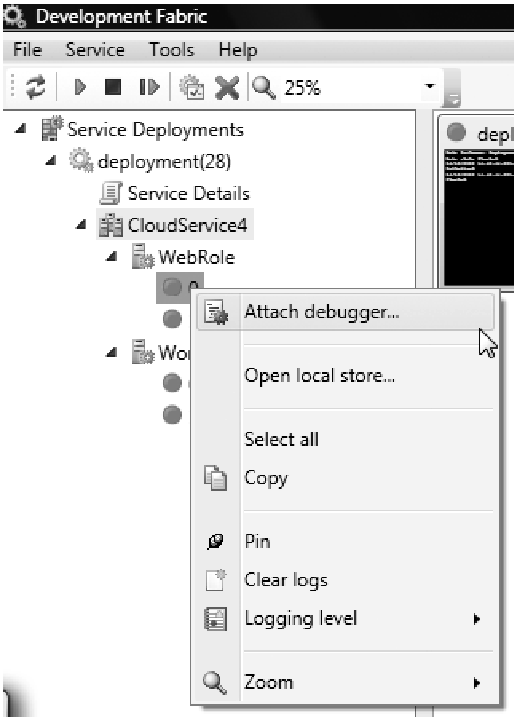
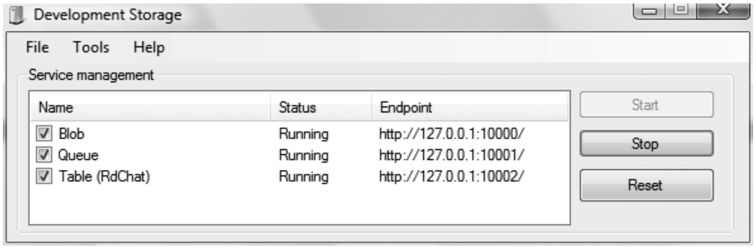


Figure 8-33. *Development fabric's Attach Debugger button*

The development fabric UI will give you the option of selecting the available debuggers on the local machine. It also allows you to set the logging levels at the service, role, and instance levels.

Development Storage

Development storage simulates the Windows Azure blobs, queues, and table storage services on your local computer over SQL Server Express 2005/2008. Development storage provides a user interface to start, stop, reset, and view the local storage services, as shown in Figure 8-34.



Name	Status	Endpoint
☑ Blob	Running	http://127.0.0.1:10000/
☑ Queue	Running	http://127.0.0.1:10001/
☑ Table (RdChat)	Running	http://127.0.0.1:10002/

Figure 8-34. *Development storage UI*

Figure 8-34 shows the name of the service, its status, and the endpoint it is listening on. From Tools Table Service Properties, you can change the database to be used by the table storage service.

■ **Note** In Windows Azure SDK 1.3, the Development Storage was renamed to Storage Emulator.

You can change the development storage to point to another database using the `DSInit.exe` tool that you saw in Table 8-2, with a `/sqlInstance` parameter.

■ **Note** Use SQL instance name without the server qualifier or use `.` (a period) for the default instance. To see all the parameters for `DSInit.exe`, go to the `bin` directory of the Windows Azure SDK installation, and run `DSInit.exe /?` from the command prompt.

Table 8-4 lists some key limitations of development storage compared to Windows Azure cloud storage.

Table 8-4. *Development Storage Limitations*

Attribute	Limitation
Authentication	Development storage only supports a single fixed developer account with a well-known authentication key.
Encryption	Development storage does not support HTTPS.
Scalability	Development storage is not designed to support a large number of concurrent clients. You should use development storage only for functional testing, not for performance or stress testing.
Flexibility	In the CTP version of Windows Azure, the development table storage required a fixed schema to be created before using the table service. The cloud table service did not have this constraint. You can use the table service directly without configuring the schema. The development storage does not require fixed schema any more. String properties in the development table cannot exceed 1,000 characters.
Size	The development blob service supports only 2GB of storage, whereas the cloud Block Blob supports 200GB of storage and Page Blob supports 1TB or storage.

In the case of authentication, the account name and account key are as follows:

Account name:
`devstoreaccount1`
Account key:
`Eby8vdM02xNOcqFlqUwJPLlmEtlCDXJ1OUzFT50uSRZ6IFsuFq2UVErCz4I6tq/K1SZFPTOtr/KBHBeksoGMGw==`

■ **Caution** Before deploying a storage service application to Windows Azure cloud, please make sure to change the development account information to your cloud account. You cannot use the development storage account to access the Windows Azure Storage service in the cloud. Usually, I create two configuration files—one for the development environment and one for the Windows Azure cloud environment—and then I swap based on the deployment destination.

Diagnostics

Logging support in the cloud is one of the biggest concerns of the developer community. With highly interactive integrated design environment (IDE) tools like Visual Studio and runtime environments like the .NET Framework, you can pinpoint problems in your code even in deployed environments when applications are running on-premise. However, the Visual Studio domain is limited to the access it has to the application's runtime environment. Visual Studio communicates with the runtime environment of the application to gather debug information of the application. The application needs to have debug symbols loaded in runtime for Visual Studio to debug. The Windows Azure development fabric has access to the local runtime environment, so you can debug your local Windows Azure application like any other .NET application by adding breakpoints.

Unfortunately, Visual Studio cannot access Windows Azure runtime directly. Once the service is deployed to Windows Azure, it is totally managed by Windows Azure, and you do not have access to its runtime. The Windows Azure team realized this and has added logging capabilities to the Windows Azure runtime and also added IntelliTrace support for Windows Azure deployments from Visual Studio Ultimate edition. The diagnostics service runs along with your role instance, collects diagnostics data as per the configuration, and can save the data to your Windows Azure Storage service if configured to do so. You can also communicate with the diagnostics service remotely from an on-premise application or configure it to persist the diagnostics data on a periodic basis. The diagnostics service supports logging of the following data types from your cloud service:

- *Windows Azure Trace logs*: These are the application logs that you dump from your application. These can be any messages emitted from your code.
- *Diagnostic Infrastructure logs*: Infrastructure logs dumped by the diagnostics service.
- *Windows event logs*: These are the Windows event logs generated on the machine on which the role instance is running.
- *Windows performance counters*: These refer to the subscriptions to the performance counters on the machine on which the role instance is running.

- *IIS logs and failed request traces*: These are the IIS logs and the IIS failed request traces generated on the Web role instance.
- *Application crash dumps*: These are the crash dumps generated when an application crashes.

The diagnostics engine aggregates all the logs together and transfers them to the appropriate storage. Windows Azure role instances are stateless and therefore you may lose locally stored logs on recycle. The diagnostics engine named "`MonAgentHost.exe`" runs on all the role instances by default. Figure 8-35 illustrates the logical architecture of the diagnostics process in one Windows Azure role instance.

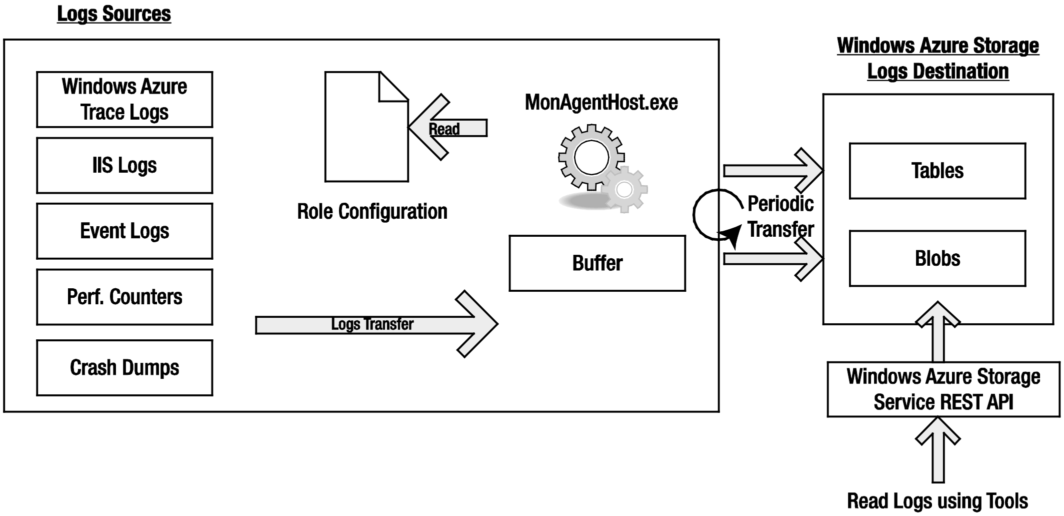


Figure 8-35. *Diagnostics in one role instance*

Table 8-5 lists the log data sources available in Windows Azure role instances and their respective destinations in the Windows Azure storage.

Table 8-5. *Development Storage Limitations*

Data Source (on Azure Role instance)	Is it enabled by default in Azure?	Destination Type (in Windows Azure Storage)	Destination Name
Windows Azure Trace Logs	Yes	Table Storage	WADLogsTable
Diagnostic Infrastructure Logs	Yes	Table Storage	WADDiagnosticInfrastructureLogsTable

Data Source (on Azure Role instance)	Is it enabled by default in Azure?	Destination Type (in Windows Azure Storage)	Destination Name
IIS Logs	Yes	Blob Storage	wad-iis-logfiles \<deployment ID> \<web role name>\ <role instance>\W3SVC1
Performance Counters	No	Table Storage	WADPerformanceCountersTable
Windows Event Logs	No	Table Storage	WADWindowsEventLogsTable
IIS Failed Request Logs	No	Blob Storage	wad-iis-failedreqlogfiles\<deployment ID>\<web role name>\<role instance>\W3SVC1
Crash Dumps	No	Blob Storage	wad-crash-dumps

You can use the diagnostics management API from outside of the Windows Azure cloud environment (e.g., on-premise) to interact with the diagnostics service on your role instance. Next, using the same API, you can perform scheduled or on-demand transfers of the diagnostics information from role instance machines to your Windows Azure Storage account. The diagnostics API is present in the `Microsoft.WindowsAzure.Diagnostics` assembly.

■ **Note** You can find more information about the Windows Azure Runtime API at the Windows Azure MSDN reference site: `http://msdn.microsoft.com/en-us/library/dd179380.aspx`.

Logging

Windows Azure Runtime API consists of a managed code library and an unmanaged code library. The complete book, *Windows Azure Platform*, Second Edition, covers the managed code library. The managed code library namespace for diagnostics is `Microsoft.WindowsAzure.Diagnostics`. Associating diagnostics with your cloud service is a three-step process:

1. Configure the trace listener.
2. Define the storage location for the diagnostics service.
3. Start the diagnostics service.

Configuring the Trace Listener

When you create a new role using the role templates template in Visual Studio, the `app.config` and `web.config` files get created automatically in the role project and it consists of a trace listener provider, as shown in Listing 8-5.

Listing 8-5. Diagnostics Trace Listener Configuration

```
<system.diagnostics>
<trace>
 <listeners>
   <add type=
"Microsoft.WindowsAzure.Diagnostics.DiagnosticMonitorTraceListener,
Microsoft.WindowsAzure.Diagnostics, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35"
name="AzureDiagnostics">
<filter type="" />
      </add>
  </listeners>
 </trace>
</system.diagnostics>
```

The `DiagnosticMonitorTraceListener` `enables` you to use the .NET Tracing API for logging within the code. You can use the `Write()` and `WriteLine()` methods of the `System.Diagnostics.Trace` class for logging from your code as shown here:

```
Trace.WriteLine("INFORMATION LOG", "Information");
Trace.WriteLine("CRITICAL LOG", "Critical");
```

Defining the Storage Location for the Diagnostics Service

In the `ServiceDefinition.csdef` and `ServiceConfiguration.cscfg` files, you have to define the diagnostics connection string pointing to the storage location of your choice (development storage or cloud storage), and custom settings like the list of PerformanceCounters. Visual Studio then automatically generates this configuration for you as shown in Listing 8-6.

Listing 8-6. Diagnostics Connection String Configuration

For development storage:

```
<ConfigurationSettings>
 <Setting name="Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString"
value="UseDevelopmentStorage=true" />
<Setting name="PerformanceCounters"
value="\Processor(_Total)\% Processor Time,\Memory\Available MBytes" />
</ConfigurationSettings>
```

For cloud storage:

```
<ConfigurationSettings>
```

```
 <Setting name="DiagnosticsConnectionString" value=
"DefaultEndpointsProtocol=https;AccountName=proazurestorage;AccountKey=[YOURKEY]"/>
<Setting name="PerformanceCounters"
value="\Processor(_Total)\% Processor Time,\Memory\Available MBytes" />
</ConfigurationSettings>
```

Starting the Diagnostics Service

Next, you have to start the diagnostics service in your role by passing in the connection string name you defined in step 2. Once started, the diagnostics monitoring service can start collecting the logged data. You can also choose to further configure the diagnostics service through the `DiagnosticMonitorConfiguration` class, as shown in Listing 8-7.

Listing 8-7. *Programmatically Changing the Diagnostics Configuration*

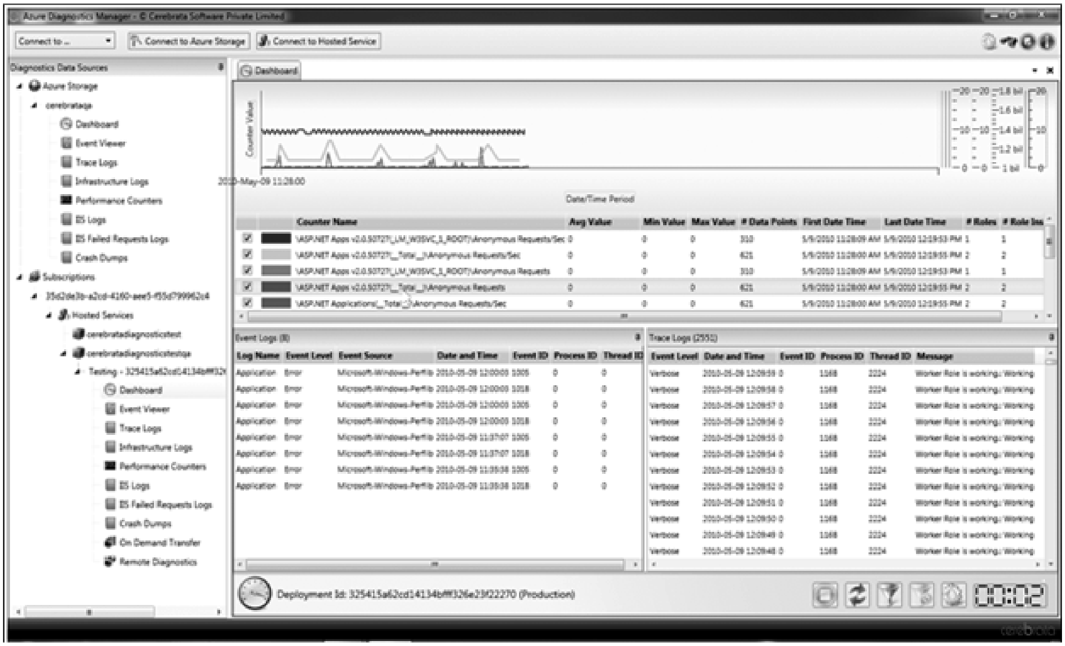
```
private void SetupDiagnostics()
        {
            Trace.WriteLine("Setting up diagnostics", "Information");

            DiagnosticMonitorConfiguration diagConfig =
DiagnosticMonitor.GetDefaultInitialConfiguration();

            // Add performance counter monitoring for configured counters
            // Run typeperf.exe /q to query the counter list
            string perfCounterString =
RoleEnvironment.GetConfigurationSettingValue("PerformanceCounters");

            if (!string.IsNullOrEmpty(perfCounterString))
            {
                IList<string> perfCounters = perfCounterString.Split(',').ToList();

                // Setup each counter specified in comma delimitered string
                foreach (string perfCounter in perfCounters)
                {
                    diagConfig.PerformanceCounters.DataSources.Add(
                        new PerformanceCounterConfiguration
                        {
                            CounterSpecifier = perfCounter,
                            SampleRate = TimeSpan.FromSeconds(5)
                        }
                        );
                }

                // Update counter information in Azure every 30 seconds
                diagConfig.PerformanceCounters.ScheduledTransferPeriod =
TimeSpan.FromMinutes(0.5);
            }

            diagConfig.DiagnosticInfrastructureLogs.ScheduledTransferPeriod =
TimeSpan.FromMinutes(0.5);
```

```
            // Specify a logging level to filter records to transfer
            diagConfig.DiagnosticInfrastructureLogs.ScheduledTransferLogLevelFilter = 
LogLevel.Verbose;

            // Set scheduled transfer interval for user's Windows Azure Logs to 5 minutes
            diagConfig.Logs.ScheduledTransferPeriod = TimeSpan.FromMinutes(5);

            diagConfig.Directories.ScheduledTransferPeriod = TimeSpan.FromMinutes(5);

            Microsoft.WindowsAzure.Diagnostics.CrashDumps.EnableCollection(true);


            //Event Logs
            // Add event collection from the Windows Event Log
            diagConfig.WindowsEventLog.DataSources.Add("System!*");
            diagConfig.WindowsEventLog.DataSources.Add("Application!*");
            diagConfig.WindowsEventLog.DataSources.Add("Security!*");
            diagConfig.WindowsEventLog.ScheduledTransferPeriod = TimeSpan.FromMinutes(5);

            // Start the diagnostic monitor with this custom configuration

DiagnosticMonitor.Start("Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString", 
diagConfig);
        }
```

In Listing 8-7, the `SetupDiagnostics()` function performs the following:

1. Initializes the Diagnostic Monitor Configuration.
2. Reads the Performance Counters listed in the `ServiceConfiguration.cscfg`.
3. Adds each Performance Counter as a data source to the Diagnostic Monitor Configuration.
4. Schedules the transfer of infrastructure and directories (e.g., IIS) logs to the Windows Azure Storage.
5. Initializes Diagnostic Infrastructure logs transfer.
6. Initializes crash dump collection and transfer.
7. Adds System, Application, and Security Event Log data sources to the Diagnostic Monitor Configuration.
8. Schedules the transfer of Event Log data.
9. Starts the Diagnostic Monitor service.

Once the logs data gets collected in the Windows Azure storage destinations, you can read it by either writing a custom application that calls the Windows Azure Storage REST API or by running a third-party tool like the Cerebrata's Azure Diagnostics Manager (`www.cerebrata.com/Products/AzureDiagnosticsManager/Default.aspx`).



		Counter Name	Avg Value	Min Value	Max Value	# Data Points	First Date Time	Last Date Time	# Roles	# Role Ins
☑	■	\ASP.NET Apps v2.0.50727(_LM_W3SVC_1_ROOT)\Anonymous Requests/Sec	0	0	0	310	5/9/2010 11:28:09 AM	5/9/2010 12:19:53 PM	1	1
☑	■	\ASP.NET Apps v2.0.50727(__Total__)\Anonymous Requests/Sec	0	0	0	621	5/9/2010 11:28:00 AM	5/9/2010 12:19:55 PM	2	2
☑	■	\ASP.NET Apps v2.0.50727(_LM_W3SVC_1_ROOT)\Anonymous Requests	0	0	0	310	5/9/2010 11:28:09 AM	5/9/2010 12:19:53 PM	1	1
☑	■	\ASP.NET Apps v2.0.50727(__Total__)\Anonymous Requests	0	0	0	621	5/9/2010 11:28:00 AM	5/9/2010 12:19:55 PM	2	2
☑	■	\ASP.NET Applications(__Total__)\Anonymous Requests/Sec	0	0	0	621	5/9/2010 11:28:00 AM	5/9/2010 12:19:55 PM	2	2

Event Logs (8)

Log Name	Event Level	Event Source	Date and Time	Event ID	Process ID	Thread ID
Application	Error	Microsoft-Windows-Perflib	2010-05-09 12:00:03	1005	0	0
Application	Error	Microsoft-Windows-Perflib	2010-05-09 12:00:03	1018	0	0
Application	Error	Microsoft-Windows-Perflib	2010-05-09 12:00:03	1005	0	0
Application	Error	Microsoft-Windows-Perflib	2010-05-09 12:00:03	1018	0	0
Application	Error	Microsoft-Windows-Perflib	2010-05-09 11:37:07	1005	0	0
Application	Error	Microsoft-Windows-Perflib	2010-05-09 11:37:07	1018	0	0
Application	Error	Microsoft-Windows-Perflib	2010-05-09 11:35:38	1005	0	0
Application	Error	Microsoft-Windows-Perflib	2010-05-09 11:35:38	1018	0	0

Trace Logs (2551)

Event Level	Date and Time	Event ID	Process ID	Thread ID	Message
Verbose	2010-05-09 12:09:59	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:58	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:57	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:56	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:55	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:54	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:53	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:52	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:51	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:50	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:49	0	1168	2224	Worker Role is working.: Working
Verbose	2010-05-09 12:09:48	0	1168	2224	Worker Role is working.: Working

Deployment Id: 325415a62cd14134bff326e23f22270 (Production)

Figure 8-36. Cerebrata Azure Diagnostics Manager

■ **Note** Microsoft has also released System Center Operations Manager (SCOM) pack for Windows Azure. You can download it here `microsoft.com/download/en/details.aspx?`

Walter Myers III has a nice series of blog articles on Windows Azure application monitoring with SCOM: `blogs.msdn.com/b/walterm/archive/2011/02/14/adding-azure-application-monitoring-to-scom-2007-r2.aspx`

■ **Tip** When designing cloud applications, it is important to design diagnostics and logs reporting right from the beginning. This will save you a lot of debugging time and help you create a high quality application.

Developing Windows Azure Services with Inter-Role Communication

In this example, you will learn to develop Windows Azure services in the local development fabric and in the cloud environment. I will also show you how to communicate between roles using the internal endpoints. You will also learn to use your own Configuration Settings in `ServiceDefinition.csdef` and `ServiceConfiguration.cscfg`.

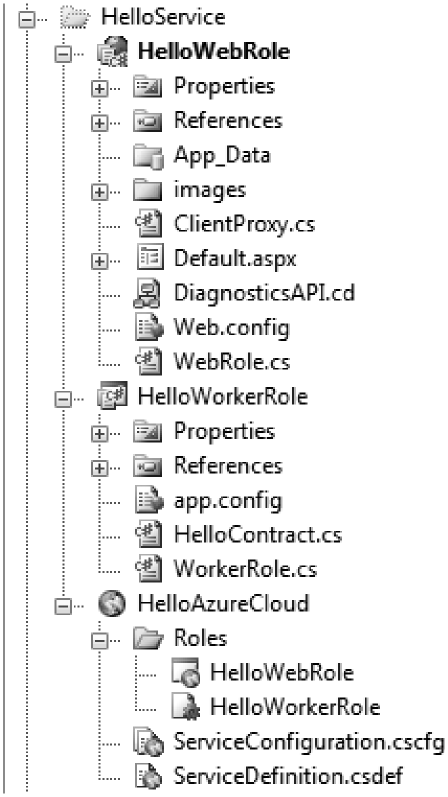
Objectives

The objectives of this example are as follows:

- Understanding inter-role communication in Windows Azure cloud services.
- Accessing local machine resources.
- Understanding the configuration settings for configuring cloud services.

Adding Diagnostics and Inter-role Communication

In this section, I will guide you through the code for adding diagnostics, configuration and inter-role communication to the Windows Azure services. Figure 8-37 shows an example folder I'll use for illustrating the process.



```
HelloService
    HelloWebRole
        Properties
        References
        App_Data
        images
        ClientProxy.cs
        Default.aspx
        DiagnosticsAPI.cd
        Web.config
        WebRole.cs
    HelloWorkerRole
        Properties
        References
        app.config
        HelloContract.cs
        WorkerRole.cs
    HelloAzureCloud
        Roles
            HelloWebRole
            HelloWorkerRole
        ServiceConfiguration.cscfg
        ServiceDefinition.csdef
```

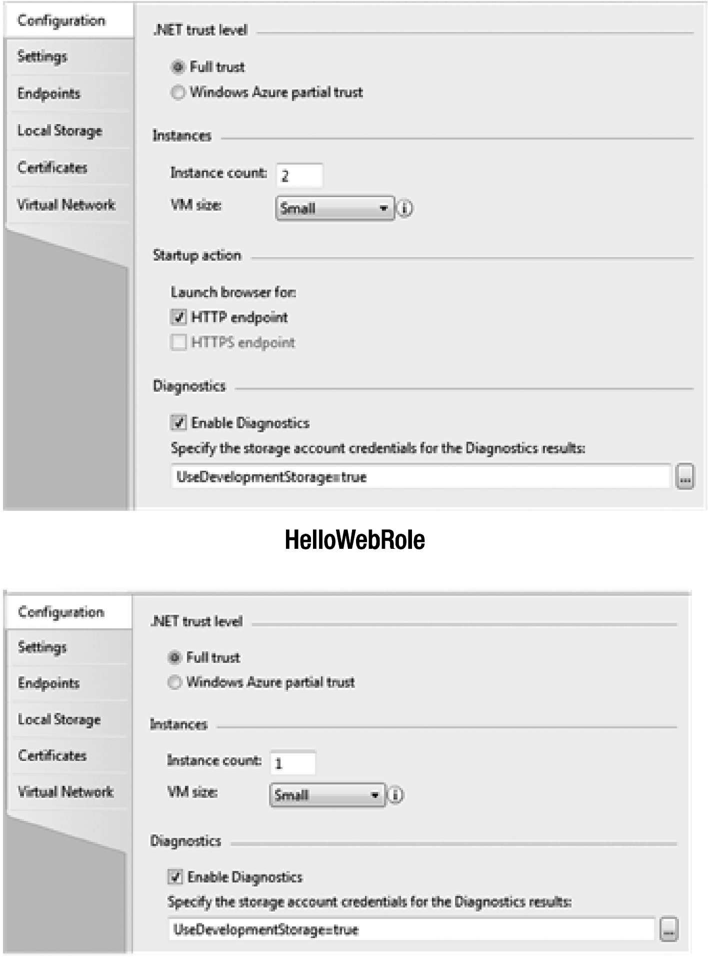
Figure 8-37. *HelloService folder*

The folder contains one Web role, one Worker role, and one cloud service project: HelloWebRole, HelloWorkerRole, and HelloAzu reCloud cloud service respectively.

Service Model

The `ServiceDefinition.csdef` and `ServiceConfiguration.cscfg` file define the service model and configuration values for the service. Figures 8-38, 8-39, 8-40, 8-41, and 8-42 illustrate the Service of the HelloAzureCloud service.

Configuration



.NET trust level

● Full trust
○ Windows Azure partial trust

Instances

Instance count: 2
VM size: Small

Startup action

Launch browser for:
☑ HTTP endpoint
□ HTTPS endpoint

Diagnostics

☑ Enable Diagnostics
Specify the storage account credentials for the Diagnostics results:
UseDevelopmentStorage=true

HelloWebRole

.NET trust level

● Full trust
○ Windows Azure partial trust

Instances

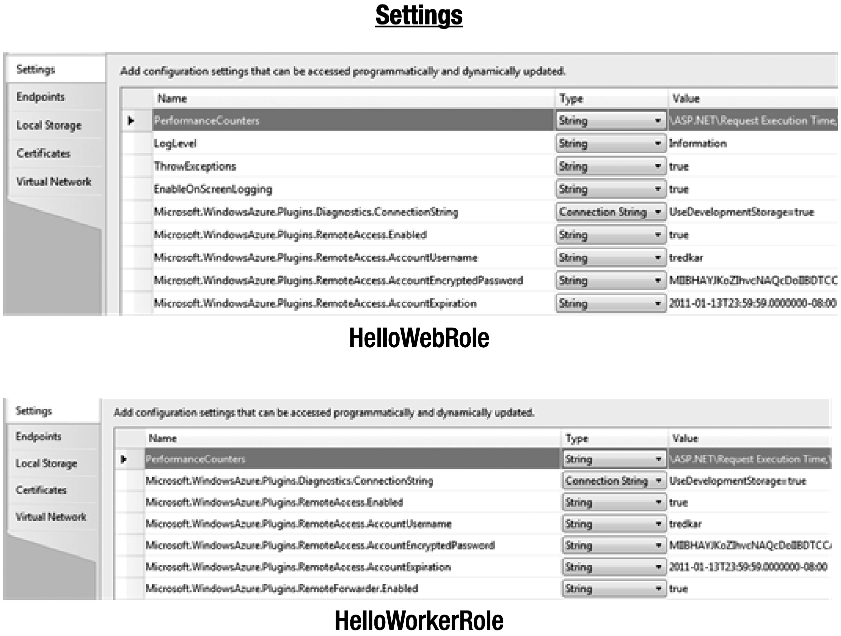
Instance count: 1
VM size: Small

Diagnostics

☑ Enable Diagnostics
Specify the storage account credentials for the Diagnostics results:
UseDevelopmentStorage=true

HelloWorkerRole

Figure 8-38. *HelloAzu reCloud – Configu ration*



Settings

Add configuration settings that can be accessed programmatically and dynamically updated.

Name	Type	Value
PerformanceCounters	String	\ASP.NET\Request Execution Time,
LogLevel	String	Information
ThrowExceptions	String	true
EnableOnScreenLogging	String	true
Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString	Connection String	UseDevelopmentStorage=true
Microsoft.WindowsAzure.Plugins.RemoteAccess.Enabled	String	true
Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountUsername	String	tredkar
Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountEncryptedPassword	String	MIIBHAYJKoZIhvcNAQcDoIIBDTCC
Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountExpiration	String	2011-01-13T23:59:59.0000000-08:00

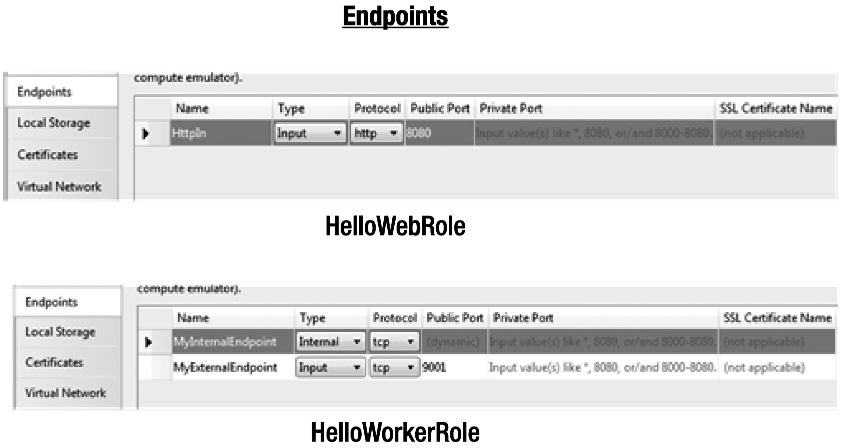
HelloWebRole

Add configuration settings that can be accessed programmatically and dynamically updated.

Name	Type	Value
PerformanceCounters	String	\ASP.NET\Request Execution Time,\
Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString	Connection String	UseDevelopmentStorage=true
Microsoft.WindowsAzure.Plugins.RemoteAccess.Enabled	String	true
Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountUsername	String	tredkar
Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountEncryptedPassword	String	MIIBHAYJKoZIhvcNAQcDoIIBDTCCA
Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountExpiration	String	2011-01-13T23:59:59.0000000-08:00
Microsoft.WindowsAzure.Plugins.RemoteForwarder.Enabled	String	true

HelloWorkerRole

Figure 8-39. *HelloAzureCloud – Settings*



Endpoints

compute emulator).

Name	Type	Protocol	Public Port	Private Port	SSL Certificate Name
HttpIn	Input	http	8080	Input value(s) like *, 8080, or/and 8000-8080.	(not applicable)

HelloWebRole

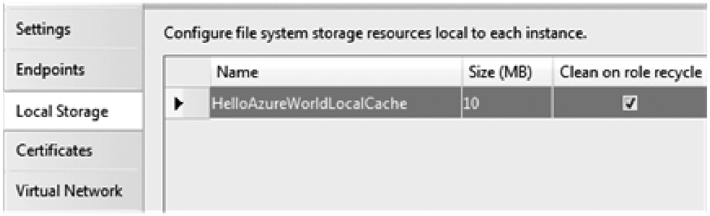
compute emulator).

Name	Type	Protocol	Public Port	Private Port	SSL Certificate Name
MyInternalEndpoint	Internal	tcp	(dynamic)	Input value(s) like *, 8080, or/and 8000-8080.	(not applicable)
MyExternalEndpoint	Input	tcp	9001	Input value(s) like *, 8080, or/and 8000-8080.	(not applicable)

HelloWorkerRole

Figure 8-40. *HelloAzureCloud – Endpoints*

Local Storage



Settings
Endpoints
Local Storage
Certificates
Virtual Network

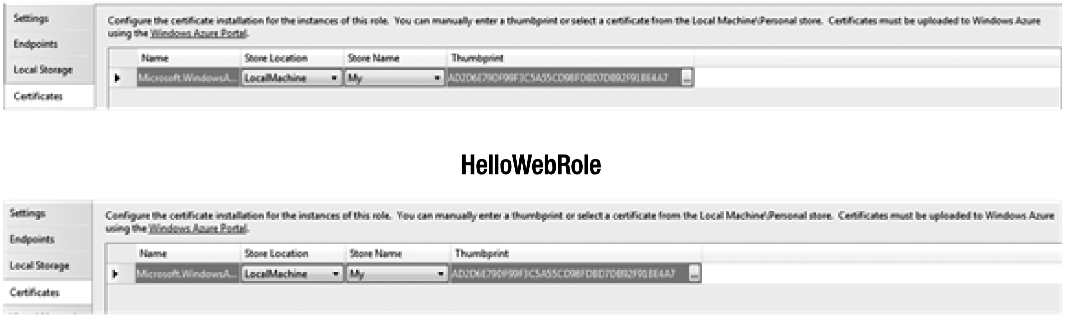
Configure file system storage resources local to each instance.

	Name	Size (MB)	Clean on role recycle
▶	HelloAzureWorldLocalCache	10	☑

HelloWebRole

Figure 8-41. *HelloAzu reCloud – Local Storage*

Certificates



Settings
Endpoints
Local Storage
Certificates

Configure the certificate installation for the instances of this role. You can manually enter a thumbprint or select a certificate from the Local Machine\Personal store. Certificates must be uploaded to Windows Azure using the Windows Azure Portal.

	Name	Store Location	Store Name	Thumbprint
▶	Microsoft.WindowsA...	LocalMachine	My	AD2D6E79DF99F3C5A55CD98FD8D7D892F91BE4A7

HelloWebRole

Settings
Endpoints
Local Storage
Certificates

Configure the certificate installation for the instances of this role. You can manually enter a thumbprint or select a certificate from the Local Machine\Personal store. Certificates must be uploaded to Windows Azure using the Windows Azure Portal.

	Name	Store Location	Store Name	Thumbprint
▶	Microsoft.WindowsA...	LocalMachine	My	AD2D6E79DF99F3C5A55CD98FD8D7D892F91BE4A7

HelloWorkerRole

Figure 8-42. *HelloAzu reCloud – Certificates*

This service model defines an external HTTP endpoint (input endpoint) for the HelloWebRole listening on port 80 and internal as well as external endpoints for the HelloWorkerRole. HelloWebRole also defines a local storage named HelloAzureWorldLocalCache with maximum size of 10MB. Both the roles define the Diagnostics and Remote Access plug-in.

The `ServiceDefinition.csdef` also defines a startup task that calls Startme.cmd, which in turn opens a command prompt.

```
<Startup>
      <Task commandLine="Startme.cmd" executionContext="elevated" taskType="simple">
      </Task>
    </Startup>
```

Worker Role

HelloWorkerRole implements two methods, `OnStart()` and `Run()`. In the `OnStart()` method, it also subscribes to the role changing event to catch any configuration changes.

■ **Note** In both the Web and the Worker roles, you need to add references to the following assemblies: `Microsoft.WindowsAzue.ServiceRuntime.dll`, `Microsoft.WindowsAzure.StorageClient.dll` and `Microsoft.WindowsAzure.Diagnostics.dll`. And you need to add the following `using` statements in code: `using Microsoft.WindowsAzure.ServiceRuntime;` and `using Microsoft.WindowsAzure.Diagnostics;`. The storage client is used for transferring logs from Windows Azure instances to the storage service.

Listing 8-8 shows the code for the `HelloWorkerRole` class.

Listing 8-8. *Hello WorkerRole*

```
public override void Run()

{
  Trace.WriteLine("HelloWorkerRole entry point called", "Information");
  var internalEndpoint =
RoleEnvironment.CurrentRoleInstance.InstanceEndpoints["MyInternalEndpoint"];
  var wcfAddress = new
Uri(String.Format("net.tcp://{0}",internalEndpoint.IPEndpoint.ToString()));
  Trace.WriteLine(wcfAddress.ToString());
  var wcfHost = new ServiceHost(typeof(HelloServiceImpl), wcfAddress);
  var binding = new NetTcpBinding(SecurityMode.None);
  wcfHost.AddServiceEndpoint(typeof(IHelloService), binding, "helloservice");
  try
  {
    wcfHost.Open();
    while (true)
    {
      Thread.Sleep(10000);
      Trace.WriteLine("Working", "Information");
    }
  }
  finally
  {
    wcfHost.Close();

  }
}

public override bool OnStart()
{
// Set the maximum number of concurrent connections
  ServicePointManager.DefaultConnectionLimit = 12;
  SetupDiagnostics();
   Trace.WriteLine("Diagnostics setup...");
            // For information on handling configuration changes
            // see the MSDN topic at http://go.microsoft.com/fwlink/?LinkId=166357.
  RoleEnvironment.Changing += RoleEnvironmentChanging;

  return base.OnStart();
  RoleEnvironment.Changing += RoleEnvironmentChanging;
  return base.OnStart();
}

private void RoleEnvironmentChanging(object sender,
RoleEnvironmentChangingEventArgs e)
{
  if (e.Changes.Any(change => change is RoleEnvironmentConfigurationSettingChange))
  e.Cancel = true;

}
```

The `OnStart()` method starts the diagnostics service (see Listing 8-7 for the `SetupDiagnostics()` method code) with a scheduled log transfer to the storage and also subscribes to the `Changing` event of the Windows Azure runtime to detect any changes to the configuration. You can use the `RoleEnvironmentChanging` event to capture the following changes:

- `RoleEnvironmentConfigurationSettingChange` to detect the changes in the service configuration.
- `RoleEnvironmentTopologyChange` to detect the changes to the role instances in the service.

In addition, you can remove a role instance from the load-balancer after the service has started by subscribing the `RoleEnvironment.StatusCheck` event and calling `SetBusy()` method on the `RoleInstanceStatusCheckEventArgs`. You can also request a recycle of the role instance on-demand by calling the `RoleEnvironment.RequestRecycle()` method. For more information on runtime API, please see the `Microsoft.WindowsAzure.ServiceRuntime` namespace in the Windows Azure SDK class documentation.

Because the diagnostics service is configured to save all the logs to the Windows Azure storage, all the `Trace.WriteLine()` statements will be sent to the storage periodically. The `Run()` method gets a reference to the internal endpoint named `MyInternalEndpoint` from the service definition and retrieves its IP address and creates a WCF service host for the `HelloServiceImpl`. Once the WCF host is opened on the internal IP address and port, any role in the service can make WCF method calls. Listing 8-9 shows the code for `IHelloService` and `HelloServiceImpl`.

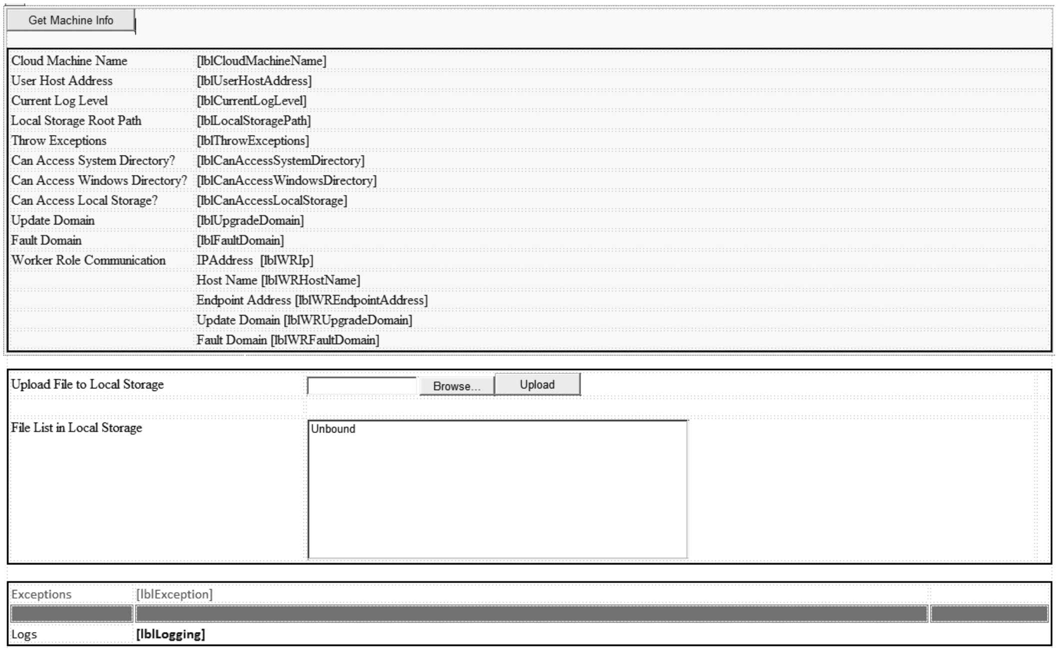
Listing 8-9. *Hello Contract*

```
[ServiceContract (Namespace="http://proazure/helloservice")]

    interface IHelloService
    {
        [OperationContract]
        string GetMyIp();
        [OperationContract]
        string GetHostName();
        [OperationContract]
        int GetUpdateDomain();
        [OperationContract]
        int GetFaultDomain();

    }
    [ServiceBehavior(AddressFilterMode=AddressFilterMode.Any)]
    public class HelloServiceImpl : IHelloService
    {


        #region IHelloService Members

        public string GetMyIp()
        {
            IPAddress[] ips = null;

            ips = Dns.GetHostAddresses(Dns.GetHostName());

            if (ips != null)
            {
                foreach (IPAddress i in ips)
                {
                    if(i.AddressFamily ==
System.Net.Sockets.AddressFamily.InterNetwork)
                     return i.ToString(); ;
                }

            }

            return "";
        }

        #endregion


        public string GetHostName()
        {
            return Dns.GetHostName();
        }

        public int GetUpdateDomain()
        {
            return RoleEnvironment.CurrentRoleInstance.UpdateDomain;

        }
        public int GetFaultDomain()
        {
            return RoleEnvironment.CurrentRoleInstance.FaultDomain;

        }

    }
```

The IHelloService interface defines four methods to retrieve the IP address, domain name of the machine, Upgrade Domain and Fault Domain. HelloServiceImpl class implements these four methods.

Web Role

The user interface for the Web role is in Default.aspx. The user interface is designed to do a few operations when you click the Get Machine Info button. Figure 8-43 illustrates the user interface design of Default.aspx page.



Cloud Machine Name	[lblCloudMachineName]
User Host Address	[lblUserHostAddress]
Current Log Level	[lblCurrentLogLevel]
Local Storage Root Path	[lblLocalStoragePath]
Throw Exceptions	[lblThrowExceptions]
Can Access System Directory?	[lblCanAccessSystemDirectory]
Can Access Windows Directory?	[lblCanAccessWindowsDirectory]
Can Access Local Storage?	[lblCanAccessLocalStorage]
Update Domain	[lblUpgradeDomain]
Fault Domain	[lblFaultDomain]
Worker Role Communication	IPAddress [lblWRIp]
	Host Name [lblWRHostName]
	Endpoint Address [lblWREndpointAddress]
	Update Domain [lblWRUpgradeDomain]
	Fault Domain [lblWRFaultDomain]

Upload File to Local Storage	
File List in Local Storage	Unbound

Exceptions	[lblException]
Logs	**[lblLogging]**

Figure 8-43. Default.aspx user interface design

When you click the Get Machine Info button, it retrieves the machine name, host address, and local storage and calls the HelloWorkerRole service through an internal endpoint. You can also upload a file to the local storage using the upload file button. All the functions use traditional .NET APIs for retrieving local file and network information of the machine. If you are developing the service from scratch, you will have to add reference to the HelloWorkRole WCF service. In the HelloWebRole project, the reference has already been added for you in the ClientProxy.cs file. Listing 8-10 shows the code for calling the HelloWorkerRole WCF service.

Listing 8-10. Call Worker Role WCF Service

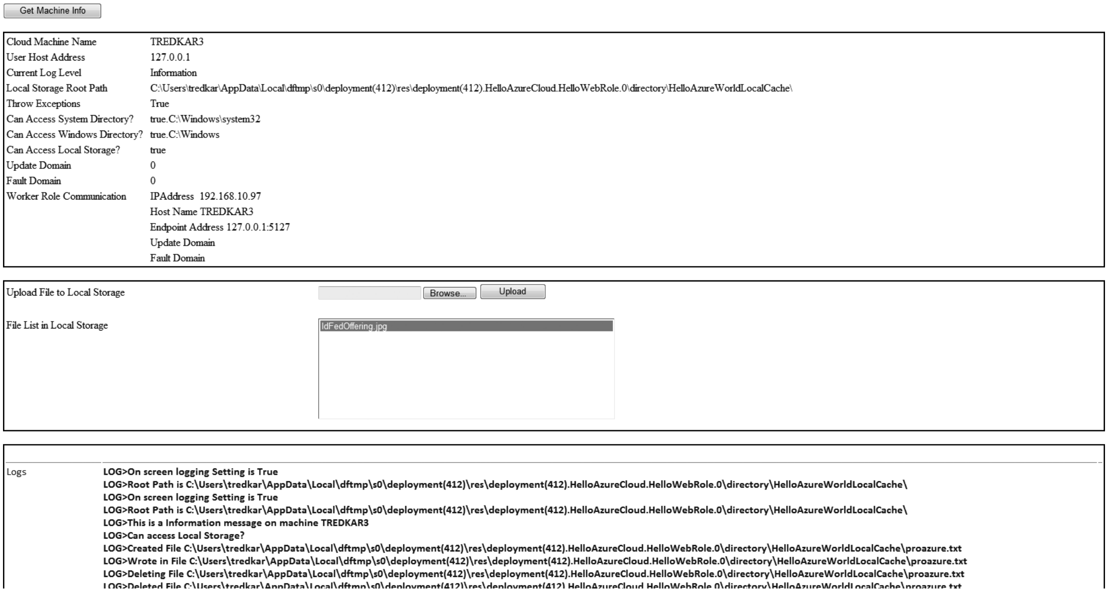
```
string wrIp =  RoleEnvironment.Roles["HelloWorkerRole"].Instances[0].
InstanceEndpoints["MyInternalEndpoint"].IPEndpoint.ToString();
lblWREndpointAddress.Text = wrIp;
var serviceAddress = new Uri(String.Format("net.tcp://{0}/{1}", wrIp, "helloservice"));
var endpointAddress = new EndpointAddress(serviceAddress);
var binding = new NetTcpBinding(SecurityMode.None);
var client = new ClientProxy(binding, endpointAddress);
lblWRHostName.Text = client.GetHostName();
lblWRIp.Text = client.GetMyIp();
lblUpgradeDomain.Text = client.GetUpdateDomain().ToString();
lblFaultDomain.Text = client.GetFaultDomain().ToString();
```

In Listing 8-10, the Web role gets reference to the internal endpoint of a Worker role instance and instantiates the `ClientProxy` object to call the `IHelloService` methods.

■ **Note** An important point to note here is that the endpoints are exposed as IP address of the instance and you still have to build your server in the form of `TcpListener`, WCF service, or HTTP service on that IP address.

Running the HelloAzureCloud Service

To build and run the solution, press F5 on the HelloAzureCloud project to start it in debug mode. Click the Get Machine Info button. Figure 8-44 illustrates the HelloAzureCloud Web role application running on the local machine.



Cloud Machine Name	TREDKAR3
User Host Address	127.0.0.1
Current Log Level	Information
Local Storage Root Path	C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(412)\res\deployment(412).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\
Throw Exceptions	True
Can Access System Directory?	true.C:\Windows\system32
Can Access Windows Directory?	true.C:\Windows
Can Access Local Storage?	true
Update Domain	0
Fault Domain	0
Worker Role Communication	IPAddress 192.168.10.97 Host Name TREDKAR3 Endpoint Address 127.0.0.1:5127 Update Domain Fault Domain

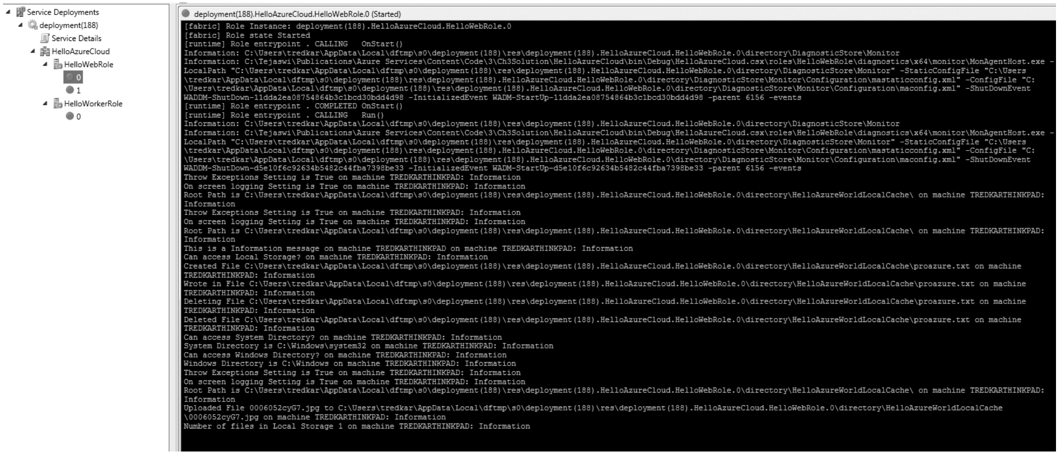
Upload File to Local Storage	
File List in Local Storage	IdFedOffering.jpg

Logs

```
LOG>On screen logging Setting is True
LOG>Root Path is C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(412)\res\deployment(412).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\
LOG>On screen logging Setting is True
LOG>Root Path is C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(412)\res\deployment(412).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\
LOG>This is a Information message on machine TREDKAR3
LOG>Can access Local Storage?
LOG>Created File C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(412)\res\deployment(412).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\proazure.txt
LOG>Wrote in File C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(412)\res\deployment(412).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\proazure.txt
LOG>Deleting File C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(412)\res\deployment(412).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\proazure.txt
LOG>Deleted File C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(412)\res\deployment(412).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\proazure.txt
```

Figure 8-44. *HelloAzureCloud on local machine*

Open the development fabric UI by clicking the development fabric icon in the system tray. Figure 8-45 shows the development fabric UI running two instances of the Web role and one instance of the Worker role.

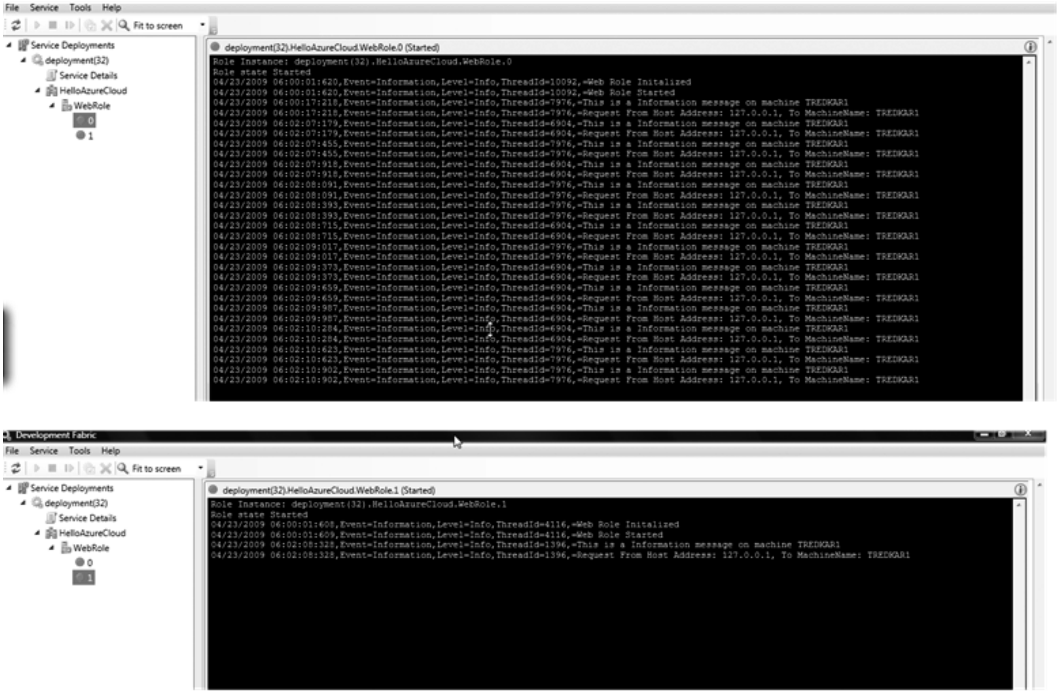


deployment(188).HelloAzureCloud.HelloWebRole.0 (Started)

```
[fabric] Role Instance: deployment(188).HelloAzureCloud.HelloWebRole.0
[fabric] Role state Started
[runtime] Role entrypoint . CALLING   OnStart()
Information: C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\DiagnosticStore\Monitor
Information: C:\Tejaswi\Publications\Azure Services\Content\Code\3\Ch3Solution\HelloAzureCloud\bin\Debug\HelloAzureCloud.csx\roles\HelloWebRole\diagnostics\x64\monitor\MonAgentHost.exe -
LocalPath "C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\DiagnosticStore\Monitor" -StaticConfigFile "C:\Users
\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\DiagnosticStore\Monitor\Configuration\mastaticconfig.xml" -ConfigFile "C:
\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\DiagnosticStore\Monitor\Configuration\maconfig.xml" -ShutDownEvent
WADDM-ShutDown-11dda2ea08754864b3c1bcd30bdd4d98 -InitializedEvent WADM-StartUp-11dda2ea08754864b3c1bcd30bdd4d98 -parent 6156 -events
[runtime] Role entrypoint . COMPLETED OnStart()
[runtime] Role entrypoint . CALLING   Run()
Information: C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\DiagnosticStore\Monitor
Information: C:\Tejaswi\Publications\Azure Services\Content\Code\3\Ch3Solution\HelloAzureCloud\bin\Debug\HelloAzureCloud.csx\roles\HelloWebRole\diagnostics\x64\monitor\MonAgentHost.exe -
LocalPath "C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\DiagnosticStore\Monitor" -StaticConfigFile "C:\Users
\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\DiagnosticStore\Monitor\Configuration\mastaticconfig.xml" -ConfigFile "C:
\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\DiagnosticStore\Monitor\Configuration\maconfig.xml" -ShutDownEvent
WADDM-ShutDown-d5e10f6c92634b5482c44fba7398be33 -InitializedEvent WADM-StartUp-d5e10f6c92634b5482c44fba7398be33 -parent 6156 -events
Throw Exceptions Setting is True on machine TREDKARTHINKPAD: Information
On screen logging Setting is True on machine TREDKARTHINKPAD: Information
Root Path is C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\ on machine TREDKARTHINKPAD:
Information
Throw Exceptions Setting is True on machine TREDKARTHINKPAD: Information
On screen logging Setting is True on machine TREDKARTHINKPAD: Information
Root Path is C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\ on machine TREDKARTHINKPAD:
Information
This is a Information message on machine TREDKARTHINKPAD on machine TREDKARTHINKPAD: Information
Can access Local Storage? on machine TREDKARTHINKPAD: Information
Created File C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\proazure.txt on machine
TREDKARTHINKPAD: Information
Wrote in File C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\proazure.txt on machine
TREDKARTHINKPAD: Information
Deleting File C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\proazure.txt on machine
TREDKARTHINKPAD: Information
Deleted File C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\proazure.txt on machine
TREDKARTHINKPAD: Information
Can access System Directory? on machine TREDKARTHINKPAD: Information
System Directory is C:\Windows\system32 on machine TREDKARTHINKPAD: Information
Can access Windows Directory? on machine TREDKARTHINKPAD: Information
Windows Directory is C:\Windows on machine TREDKARTHINKPAD: Information
Throw Exceptions Setting is True on machine TREDKARTHINKPAD: Information
On screen logging Setting is True on machine TREDKARTHINKPAD: Information
Root Path is C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache\ on machine TREDKARTHINKPAD:
Information
Uploaded File 0006052cyG7.jpg to C:\Users\tredkar\AppData\Local\dftmp\s0\deployment(188)\res\deployment(188).HelloAzureCloud.HelloWebRole.0\directory\HelloAzureWorldLocalCache
\0006052cyG7.jpg on machine TREDKARTHINKPAD: Information
Number of files in Local Storage 1 on machine TREDKARTHINKPAD: Information
```

Figure 8-45. *HelloAzureCloud development fabric two instances*

The information is logged either in the console of instance 0 or instance 1 depending on where the load balancer sends the request. If you click the Get Machine Info button very quickly, you will see that the request gets load balanced across both the instances. Figure 8-46 shows the load-balanced requests across two instances of the Web role application.



```
Role Instance: deployment(32).HelloAzureCloud.WebRole.0
Role state Started
04/23/2009 06:00:01:620,Event=Information,Level=Info,ThreadId=10092,=Web Role Initalized
04/23/2009 06:00:01:620,Event=Information,Level=Info,ThreadId=10092,=Web Role Started
04/23/2009 06:00:17:218,Event=Information,Level=Info,ThreadId=7976,=This is a Information message on machine TREDKAR1
04/23/2009 06:00:17:218,Event=Information,Level=Info,ThreadId=7976,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:07:179,Event=Information,Level=Info,ThreadId=6904,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:07:179,Event=Information,Level=Info,ThreadId=6904,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:07:455,Event=Information,Level=Info,ThreadId=7976,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:07:455,Event=Information,Level=Info,ThreadId=7976,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:07:918,Event=Information,Level=Info,ThreadId=6904,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:07:918,Event=Information,Level=Info,ThreadId=6904,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:08:091,Event=Information,Level=Info,ThreadId=7976,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:08:091,Event=Information,Level=Info,ThreadId=7976,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:08:393,Event=Information,Level=Info,ThreadId=7976,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:08:393,Event=Information,Level=Info,ThreadId=7976,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:08:715,Event=Information,Level=Info,ThreadId=6904,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:08:715,Event=Information,Level=Info,ThreadId=6904,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:09:017,Event=Information,Level=Info,ThreadId=7976,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:09:017,Event=Information,Level=Info,ThreadId=7976,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:09:373,Event=Information,Level=Info,ThreadId=6904,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:09:373,Event=Information,Level=Info,ThreadId=6904,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:09:659,Event=Information,Level=Info,ThreadId=6904,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:09:659,Event=Information,Level=Info,ThreadId=6904,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:09:987,Event=Information,Level=Info,ThreadId=6904,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:09:987,Event=Information,Level=Info,ThreadId=6904,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:10:284,Event=Information,Level=Info,ThreadId=6904,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:10:284,Event=Information,Level=Info,ThreadId=6904,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:10:623,Event=Information,Level=Info,ThreadId=7976,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:10:623,Event=Information,Level=Info,ThreadId=7976,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
04/23/2009 06:02:10:902,Event=Information,Level=Info,ThreadId=7976,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:10:902,Event=Information,Level=Info,ThreadId=7976,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
```

```
Role Instance: deployment(32).HelloAzureCloud.WebRole.1
Role state Started
04/23/2009 06:00:01:608,Event=Information,Level=Info,ThreadId=4116,=Web Role Initalized
04/23/2009 06:00:01:609,Event=Information,Level=Info,ThreadId=4116,=Web Role Started
04/23/2009 06:02:08:328,Event=Information,Level=Info,ThreadId=1396,=This is a Information message on machine TREDKAR1
04/23/2009 06:02:08:328,Event=Information,Level=Info,ThreadId=1396,=Request From Host Address: 127.0.0.1, To MachineName: TREDKAR1
```

Figure 8-46. *Load Balance across two instances of HelloAzureCloud service*

In Figure 8-46, observe the logs in the consoles of both the instances of HelloAzureCloud Web roles.

Now that you have tested the cloud service in the development fabric, you can deploy it in the Windows Azure cloud. When you deploy the application to the cloud, the consoles that you see in the development fabric are not available to visually view the logs. So, let's see how you can access and view these logs in the cloud.

Publishing to Windows Azure Cloud

For publishing the HellAzureCloud service to Windows Azure, you should first provision a service. The following steps outline how:

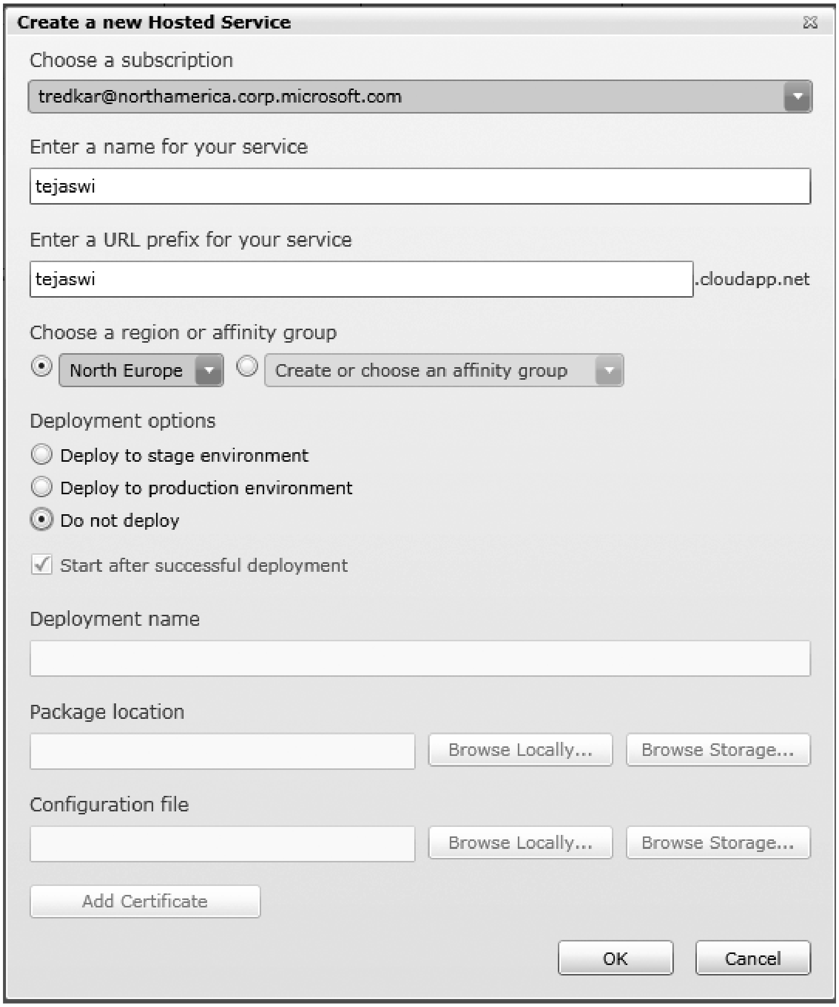
1. Log in to `https://windows.azure.com`.
2. Go to the Hosted Services section.
3. Click New Hosted Service from the top menu.
4. Enter a name for your service.
5. Choose a URL prefix for your service.
6. Choose a region closest to your location.

7. In the Deployment Options, select “Do not deploy,” because in this exercise you will deploy from the Visual Studio directly.

8. Click OK.

9. Add a certificate under the certificates folder.

10. Repeat these steps to create a new storage service.



Create a new Hosted Service

Choose a subscription: tredkar@northamerica.corp.microsoft.com

Enter a name for your service: tejaswi

Enter a URL prefix for your service: tejaswi .cloudapp.net

Choose a region or affinity group: ● North Europe ○ Create or choose an affinity group

Deployment options

○ Deploy to stage environment
○ Deploy to production environment
● Do not deploy

☑ Start after successful deployment

Deployment name

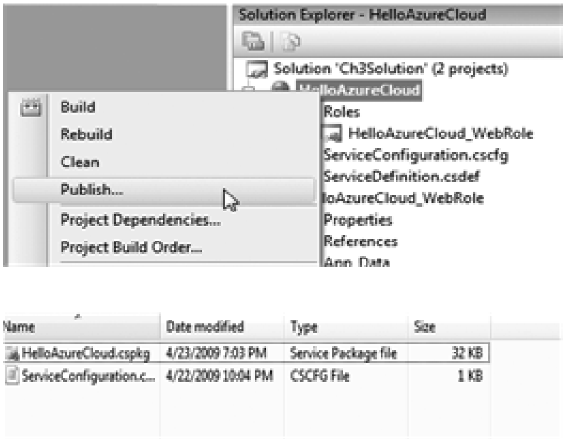
Package location

Configuration file

Figure 8-47. *Create a new Hosted Service*

■ **Note** For information on creating your own X.509 certificate, visit `http://msdn.microsoft.com/en-us/library/gg432987.aspx`.

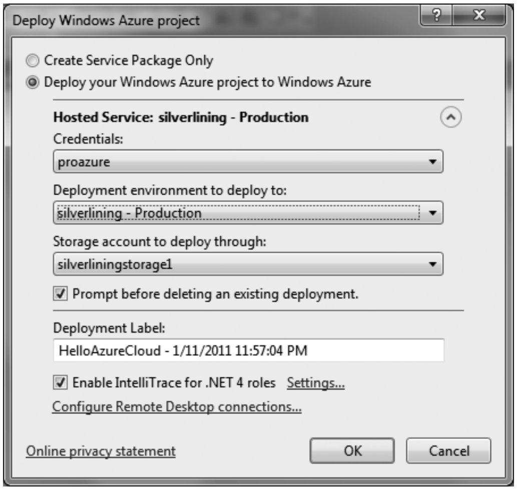
To deploy HelloAzureCloud service to Windows Azure, right-click on the HelloAzureCloud project in Visual Studio, and select Publish to create the service package `HelloAzureCloud.cspkg`, as shown in the Figure 8-48.



Name	Date modified	Type	Size
HelloAzureCloud.cspkg	4/23/2009 7:03 PM	Service Package file	32 KB
ServiceConfiguration.c...	4/22/2009 10:04 PM	CSCFG File	1 KB

Figure 8-48. *Publish to Windows Azure*

Open the Deploy Windows Azure Project dialog box, as shown in Figure 8-49.


Deploy Windows Azure project

○ Create Service Package Only
● Deploy your Windows Azure project to Windows Azure

Hosted Service: silverlining - Production

Credentials: proazure

Deployment environment to deploy to: silverlining - Production

Storage account to deploy through: silverliningstorage1

☑ Prompt before deleting an existing deployment.

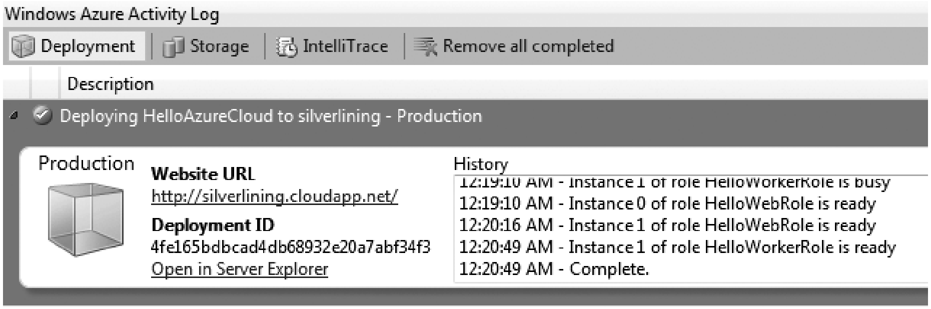
Deployment Label: HelloAzureCloud - 1/11/2011 11:57:04 PM

☑ Enable IntelliTrace for .NET 4 roles

Figure 8-49. *Deploy Windows Azure Project Windows*

Here, follow these steps:

1. Select the certificate your service management certificate you created in the previous section.
2. Select the storage location for storing the uploaded package.
3. Check Enable IntelliTrace (optional).
4. Configure Remote Desktop Connections (optional).
5. Click OK to start the Windows Azure Activity Log.
6. The windows will show completed status on successful deployment of the project, as shown in Figure 8-50.



Deploying HelloAzureCloud to silverlining - Production

Website URL
http://silverlining.cloudapp.net/

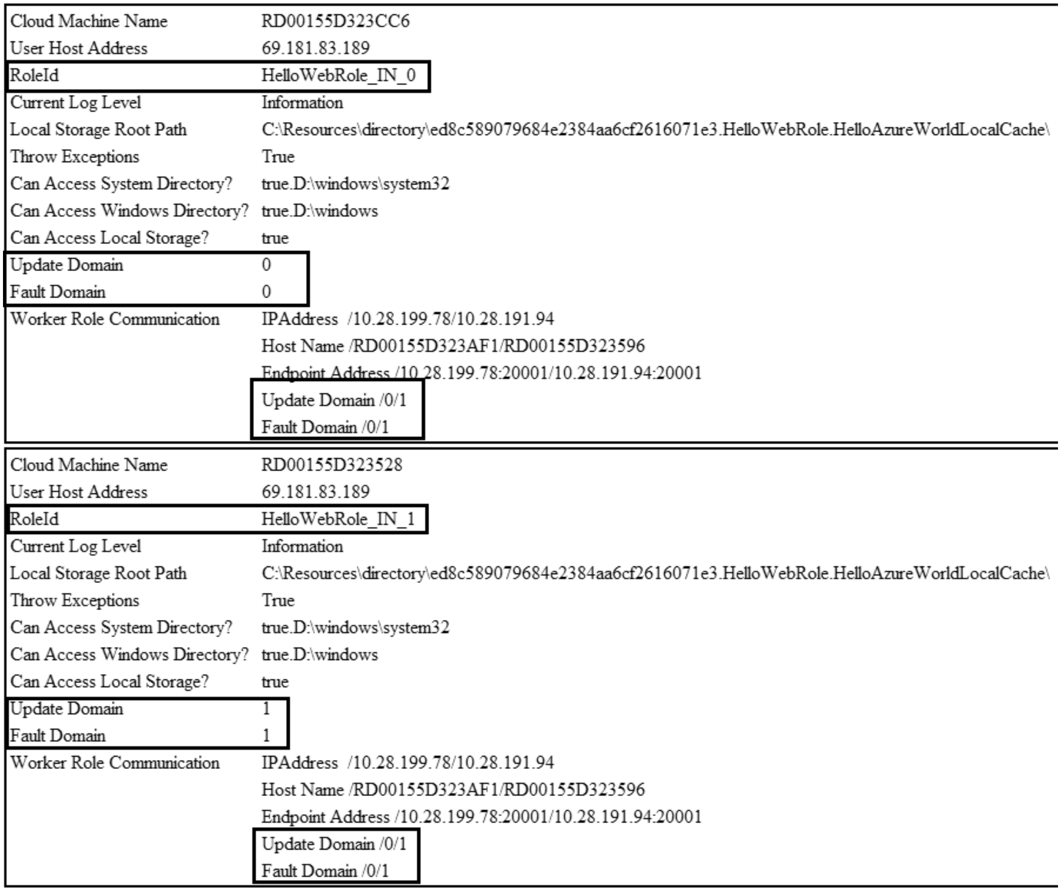
Deployment ID
4fe165bdbcad4db68932e20a7abf34f3

```
12:19:10 AM - Instance 1 of role HelloWorkerRole is busy
12:19:10 AM - Instance 0 of role HelloWebRole is ready
12:20:16 AM - Instance 1 of role HelloWebRole is ready
12:20:49 AM - Instance 1 of role HelloWorkerRole is ready
12:20:49 AM - Complete.
```

Figure 8-50. *Windows Azure Activity Log*

For viewing the application, you can point to the URL of the service and the appropriate HTTP port corresponding to the HttpIn input endpoint you created for the web role. In the previous example, the Web role's HTTP input endpoint was configured to listen on TCP port 8080. There the appropriate URL for the HelloAzureCloud service is http://silverlining.cloudapp.net:8080/Default.aspx.

Figure 8-51 illustrates the HelloAzureCloud service running in Windows Azure.



Cloud Machine Name	RD00155D323CC6
User Host Address	69.181.83.189
RoleId	HelloWebRole_IN_0
Current Log Level	Information
Local Storage Root Path	C:\Resources\directory\ed8c589079684e2384aa6cf2616071e3.HelloWebRole.HelloAzureWorldLocalCache\
Throw Exceptions	True
Can Access System Directory?	true.D:\windows\system32
Can Access Windows Directory?	true.D:\windows
Can Access Local Storage?	true
Update Domain	0
Fault Domain	0
Worker Role Communication	IPAddress /10.28.199.78/10.28.191.94 Host Name /RD00155D323AF1/RD00155D323596 Endpoint Address /10.28.199.78:20001/10.28.191.94:20001 Update Domain /0/1 Fault Domain /0/1

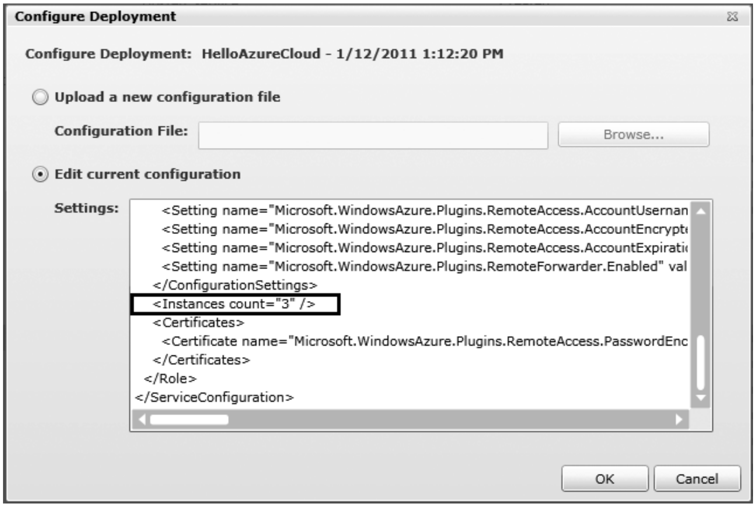
Cloud Machine Name	RD00155D323528
User Host Address	69.181.83.189
RoleId	HelloWebRole_IN_1
Current Log Level	Information
Local Storage Root Path	C:\Resources\directory\ed8c589079684e2384aa6cf2616071e3.HelloWebRole.HelloAzureWorldLocalCache\
Throw Exceptions	True
Can Access System Directory?	true.D:\windows\system32
Can Access Windows Directory?	true.D:\windows
Can Access Local Storage?	true
Update Domain	1
Fault Domain	1
Worker Role Communication	IPAddress /10.28.199.78/10.28.191.94 Host Name /RD00155D323AF1/RD00155D323596 Endpoint Address /10.28.199.78:20001/10.28.191.94:20001 Update Domain /0/1 Fault Domain /0/1

Figure 8-51. *HelloAzureCloud running in Windows Azure*

Here are a few things to observe when running the HelloAzureCloud service in Windows Azure:

1. The IPAddress and the machine name changes in Windows Azure.
2. The format of the RoleIds id is different than running in development fabric.
3. Two Upgrade Domains and two Fault Domains are utilized.
4. If you refresh the page often or start another browser, you will see the changes in the Web role information because the request is load-balanced to the second instance. The second instance will show the second Upgrade Domain and Fault Domain.

Next, select the deployed service and click the Configure button from the top menu. On the Configure Deployment screen, select the Edit Configuration option and change the number of instances of the Worker role from 2 to 3 as shown in Figure 8-52. While the roles are updating, continue to interact with the HelloAzureCloud application through `Default.aspx`. Observe that even if the configuration changes at runtime, the existing role instances continue to run unaffected.

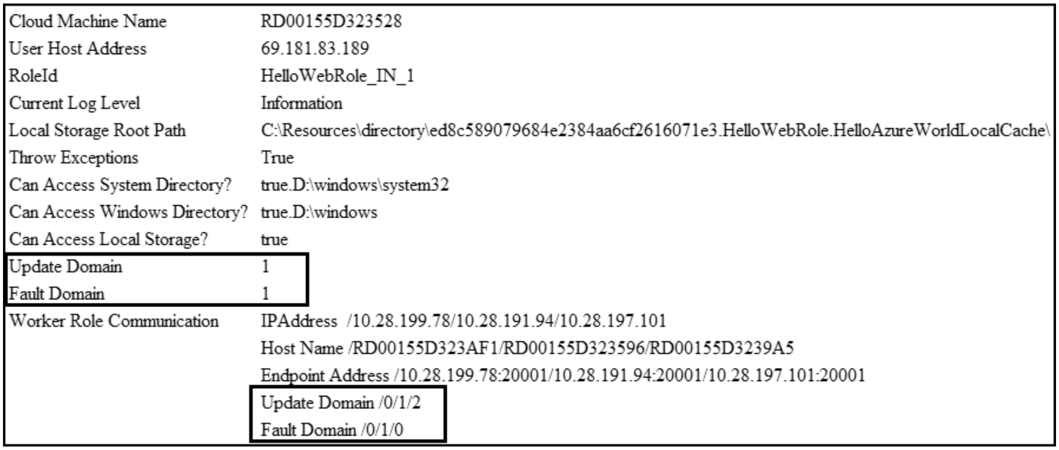

Configure Deployment
Configure Deployment: HelloAzureCloud - 1/12/2011 1:12:20 PM
○ Upload a new configuration file
Configuration File:
Browse...
● Edit current configuration
Settings:
OK
Cancel

```
    <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountUsernan
    <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountEncrypt
    <Setting name="Microsoft.WindowsAzure.Plugins.RemoteAccess.AccountExpirati
    <Setting name="Microsoft.WindowsAzure.Plugins.RemoteForwarder.Enabled" val
   </ConfigurationSettings>
   <Instances count="3" />
   <Certificates>
    <Certificate name="Microsoft.WindowsAzure.Plugins.RemoteAccess.PasswordEnc
   </Certificates>
  </Role>
</ServiceConfiguration>
```

Figure 8-52. Edit the configuration

After all the instances get into the Ready state, go back to the HelloAzureCloud application and click the GetMachineInfo button. You will observe that the Windows Azure Fabric Controller has placed the third worker role instance in a third Upgrade Domain (2) but in one of the two existing Fault Domains (0 or 1) as shown in Figure 8-53. It is important to know these settings when you are planning for high-availability and maintenance. You can also specify the number of Upgrade Domains you want to you in the Service Definition file:

```
<ServiceDefinition name="<service-name>" upgradeDomainCount="<number-of-upgrade-domains>">
```



Cloud Machine Name	RD00155D323528
User Host Address	69.181.83.189
RoleId	HelloWebRole_IN_1
Current Log Level	Information
Local Storage Root Path	C:\Resources\directory\ed8c589079684e2384aa6cf2616071e3.HelloWebRole.HelloAzureWorldLocalCache\
Throw Exceptions	True
Can Access System Directory?	true.D:\windows\system32
Can Access Windows Directory?	true.D:\windows
Can Access Local Storage?	true
Update Domain	1
Fault Domain	1
Worker Role Communication	IPAddress /10.28.199.78/10.28.191.94/10.28.197.101 Host Name /RD00155D323AF1/RD00155D323596/RD00155D3239A5 Endpoint Address /10.28.199.78:20001/10.28.191.94:20001/10.28.197.101:20001 Update Domain /0/1/2 Fault Domain /0/1/0

Figure 8-53. The Upgrade Domain and Fault Domain

For upgrading your service in-place, follow these steps:

1. Log in to Windows Azure Management Portal.
2. Go to Hosted Services.
3. Select the service you want to upgrade.
4. Click on the Upgrade button from the top menu.
5. Enter the new package and configuration information.
6. In the Upgrade Mode, specify Automatic for upgrading all Upgrade Domains at the same time and Manual for upgrading one Upgrade Domain at a time.
7. Click OK and proceed with a Manual Upgrade when the service state changes to "Ready for Manual Upgrade."

■ **Note** For more information on in-place upgrades, please visit `http://msdn.microsoft.com/en-us/library/ee517255.aspx`.

Viewing IntelliTrace Logs

When you deployed the HelloAzureCloud service to Windows Azure, you selected the checkbox for enabling IntelliTrace. I have found IntelliTrace logs handy in cases where the deployment did not go well due to missing dependencies or incorrect configuration settings. Such mistakes are difficult to catch in cloud environments.

For viewing the IntelliTrace logs of the deployment, go to Server Explorer, right-click on the cloud service instances and select View IntelliTrace logs as shown in Figure 8-54.

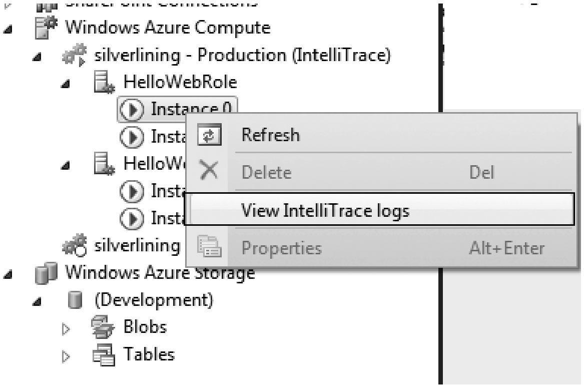
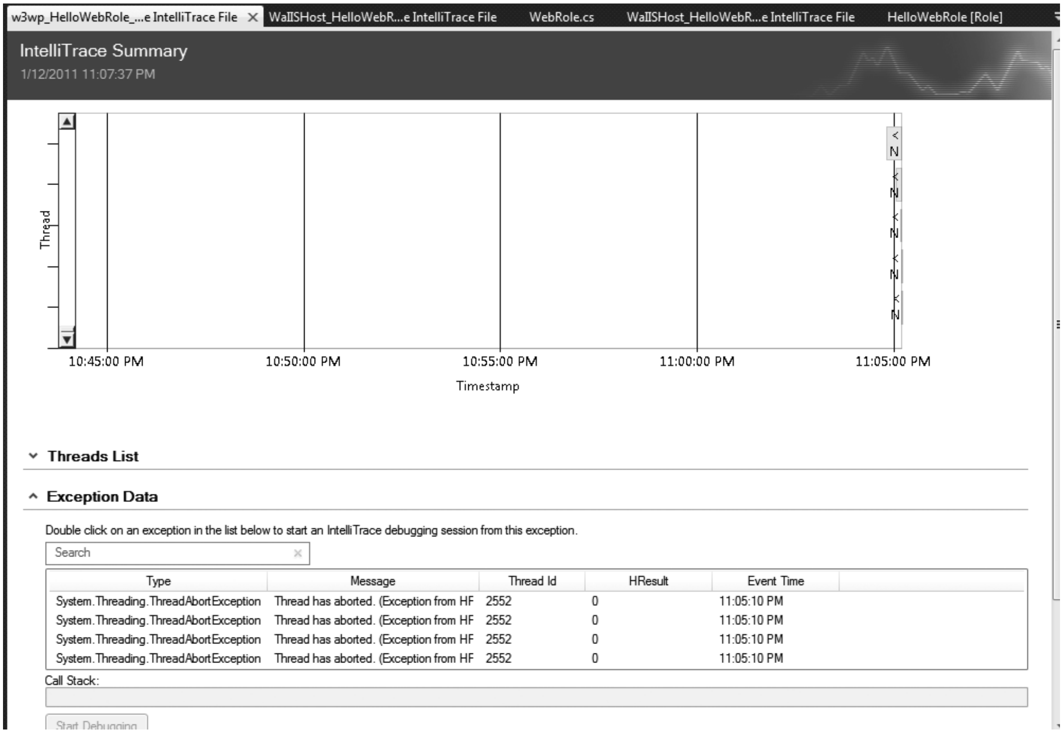


__Figure 8-54.__ Download IntellitTace Logs

This action will start downloading IntelliTrace logs from the storage service and open the summary in Visual Studio as shown in Figure 8-55.



Double click on an exception in the list below to start an IntelliTrace debugging session from this exception.

Type	Message	Thread Id	HResult	Event Time
System.Threading.ThreadAbortException	Thread has aborted. (Exception from HF	2552	0	11:05:10 PM
System.Threading.ThreadAbortException	Thread has aborted. (Exception from HF	2552	0	11:05:10 PM
System.Threading.ThreadAbortException	Thread has aborted. (Exception from HF	2552	0	11:05:10 PM
System.Threading.ThreadAbortException	Thread has aborted. (Exception from HF	2552	0	11:05:10 PM

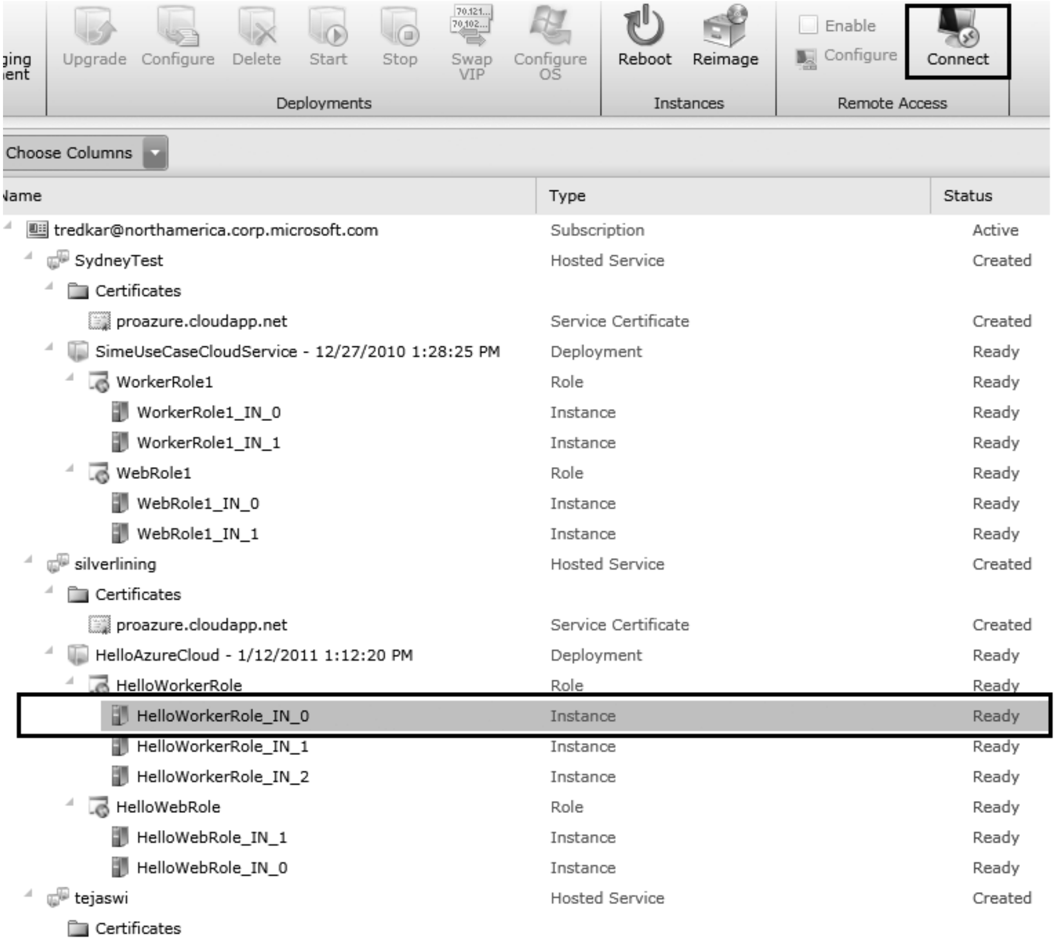
Figure 8-55. *IntelliTrace Logs*

The IntelliTrace log contains the recording of the deployment. The recording summary includes a time chart, exception data, threads list system information, and modules. With this exception and stack trace data, you can pinpoint certain type of errors very quickly.

Connecting with Remote Desktop Connection

When you published the cloud service to Windows Azure from Visual Studio, you had the option for enabling Remote Access. If you enabled remote access, you can log in to the cloud service instances. Enable remote access as outlined in the following steps:

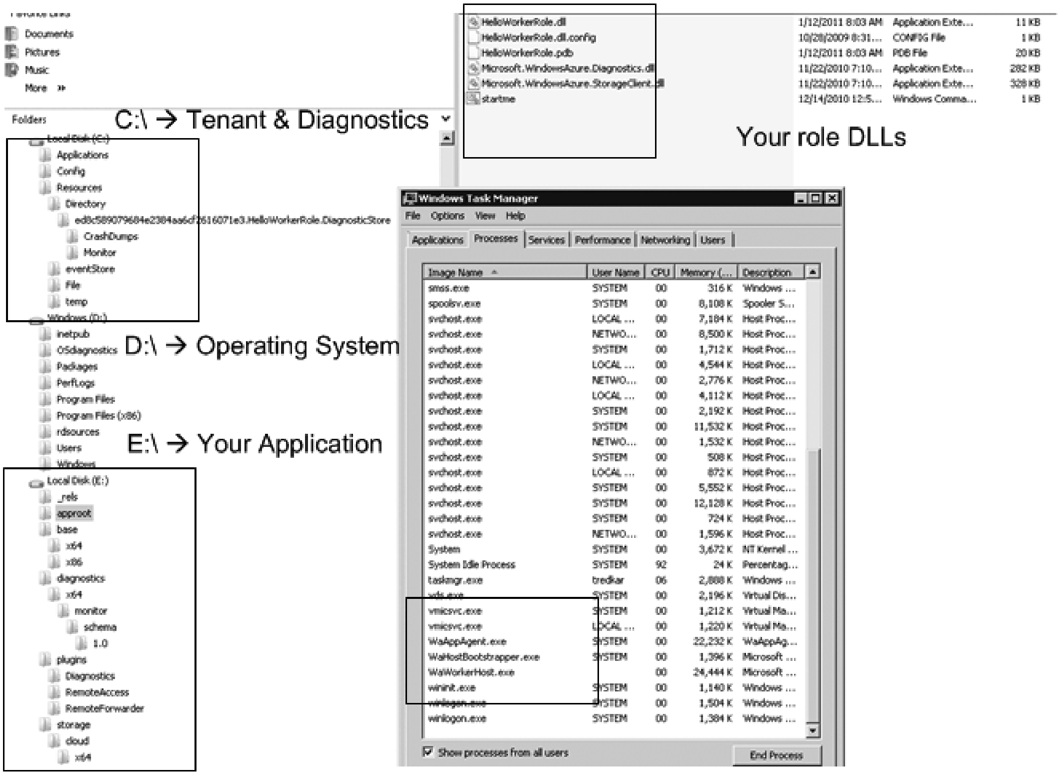
1. Go to the Hosted Services page on the Management portal.
2. Select a specific instance and click Connect from the top menu, as shown in Figure 8-56. If you recall, we have added a startup task in the Worker Role service definition for opening a command window, to check whether the command windows was opened during startup, select the HelloWorkerRole.



Name	Type	Status
tredkar@northamerica.corp.microsoft.com	Subscription	Active
SydneyTest	Hosted Service	Created
Certificates		
proazure.cloudapp.net	Service Certificate	Created
SimeUseCaseCloudService - 12/27/2010 1:28:25 PM	Deployment	Ready
WorkerRole1	Role	Ready
WorkerRole1_IN_0	Instance	Ready
WorkerRole1_IN_1	Instance	Ready
WebRole1	Role	Ready
WebRole1_IN_0	Instance	Ready
WebRole1_IN_1	Instance	Ready
silverlining	Hosted Service	Created
Certificates		
proazure.cloudapp.net	Service Certificate	Created
HelloAzureCloud - 1/12/2011 1:12:20 PM	Deployment	Ready
HelloWorkerRole	Role	Ready
HelloWorkerRole_IN_0	Instance	Ready
HelloWorkerRole_IN_1	Instance	Ready
HelloWorkerRole_IN_2	Instance	Ready
HelloWebRole	Role	Ready
HelloWebRole_IN_1	Instance	Ready
HelloWebRole_IN_0	Instance	Ready
tejaswi	Hosted Service	Created
Certificates		

Figure 8-56. *Remote Desktop connect*

After connection, experiencing Windows Azure instances using Remote Desktop instances is similar to connecting to any other Windows Server 2008. Once you remote log in, observe the directory structure and also open the task manager to see the processes running. Figure 8-57 illustrates the overall structure of the Worker Role instance. Similar structure exists for the Web roles.



C:\ → Tenant & Diagnostics

D:\ → Operating System

E:\ → Your Application

Your role DLLs

HelloWorkerRole.dll	1/12/2011 8:03 AM	Application Exte...	11 KB
HelloWorkerRole.dll.config	10/28/2009 8:31...	CONFIG File	1 KB
HelloWorkerRole.pdb	1/12/2011 8:03 AM	PDB File	20 KB
Microsoft.WindowsAzure.Diagnostics.dll	11/22/2010 7:10...	Application Exte...	282 KB
Microsoft.WindowsAzure.StorageClient.dll	11/22/2010 7:10...	Application Exte...	328 KB
startme	12/14/2010 12:5...	Windows Comma...	1 KB

Image Name	User Name	CPU	Memory (...	Description
smss.exe	SYSTEM	00	316 K	Windows ...
spoolsv.exe	SYSTEM	00	8,108 K	Spooler S...
svchost.exe	LOCAL ...	00	7,184 K	Host Proc...
svchost.exe	NETWO...	00	8,500 K	Host Proc...
svchost.exe	SYSTEM	00	1,712 K	Host Proc...
svchost.exe	LOCAL ...	00	4,544 K	Host Proc...
svchost.exe	NETWO...	00	2,776 K	Host Proc...
svchost.exe	LOCAL ...	00	4,112 K	Host Proc...
svchost.exe	SYSTEM	00	2,192 K	Host Proc...
svchost.exe	SYSTEM	00	11,532 K	Host Proc...
svchost.exe	NETWO...	00	1,532 K	Host Proc...
svchost.exe	SYSTEM	00	508 K	Host Proc...
svchost.exe	LOCAL ...	00	872 K	Host Proc...
svchost.exe	SYSTEM	00	5,552 K	Host Proc...
svchost.exe	SYSTEM	00	12,128 K	Host Proc...
svchost.exe	SYSTEM	00	724 K	Host Proc...
svchost.exe	NETWO...	00	1,596 K	Host Proc...
System	SYSTEM	00	3,672 K	NT Kernel ...
System Idle Process	SYSTEM	92	24 K	Percentag...
taskmgr.exe	tredkar	06	2,888 K	Windows ...
vds.exe	SYSTEM	00	2,196 K	Virtual Dis...
vmicsvc.exe	SYSTEM	00	1,212 K	Virtual Ma...
vmicsvc.exe	LOCAL ...	00	1,220 K	Virtual Ma...
WaAppAgent.exe	SYSTEM	00	22,232 K	WaAppAg...
WaHostBootstrapper.exe	SYSTEM	00	1,396 K	Microsoft ...
WaWorkerHost.exe		00	24,444 K	Microsoft ...
wininit.exe	SYSTEM	00	1,140 K	Windows ...
winlogon.exe	SYSTEM	00	1,504 K	Windows ...
winlogon.exe	SYSTEM	00	1,384 K	Windows ...

☑ Show processes from all users

Figure 8-57. Directory structure of a Worker role

■ **Tip** Any modifications to the instance done manually may be erased during a recycle automatically done by the Fabric Controller. Therefore, don't consider any modifications to the instances permanent. Consider each instance as a stateless and maintain all the state in your application package or externally in Windows Azure Storage.

Geo-location

Windows Azure is already available in six data centers around the world, and going forward, Microsoft plans to expand into more data centers. In today's enterprise, as well as consumer applications, the common pain point is to design a globally available service. The service needs to be physically deployed into data centers around the world for business continuity, performance, network latency, compliance, or geopolitical reasons. For example, in one project I had the responsibility for architecting a global deployment of a business critical application for a Fortune 100 company. Even though I did not need to travel around the world, I had to plan and coordinate deployment efforts around five data centers across the world. The effort took six months of rigorous planning and coordination. With geo-location support in Windows Azure, you can choose the geo-location of the storage and the compute at the time of deployment, so you don't need to deploy hardware and software physically in global locations. Table 8-6 lists some of the common geo-location advantages.

Table 8-6. Geolocation Advantages

Advantage	Rationale
Business Continuity and Planning	With geo-location features, enterprise data can be replicated across multiple data centers around the world as an insurance shield from natural and political disasters.
Performance and Network Latency	One of the architectural tenets and best practices of cloud services is keeping data close to the application for optimizing performance and end user experience. With geo-location support, a cloud service application can be run in close proximity to the data for improved performance.
Compliance	Compliance laws are different in different countries. Multinational organizations have to deal with compliance regulations in all the countries that they do business in. With Windows Azure, companies can now move data closer to the country offices for adhering to the country specific compliance regulations.
Geopolitical Requirements	Some countries pose restrictions and constraints on enterprises in where they can store enterprise data. Geo-location features can help enterprises better align with such geopolitical requirements.

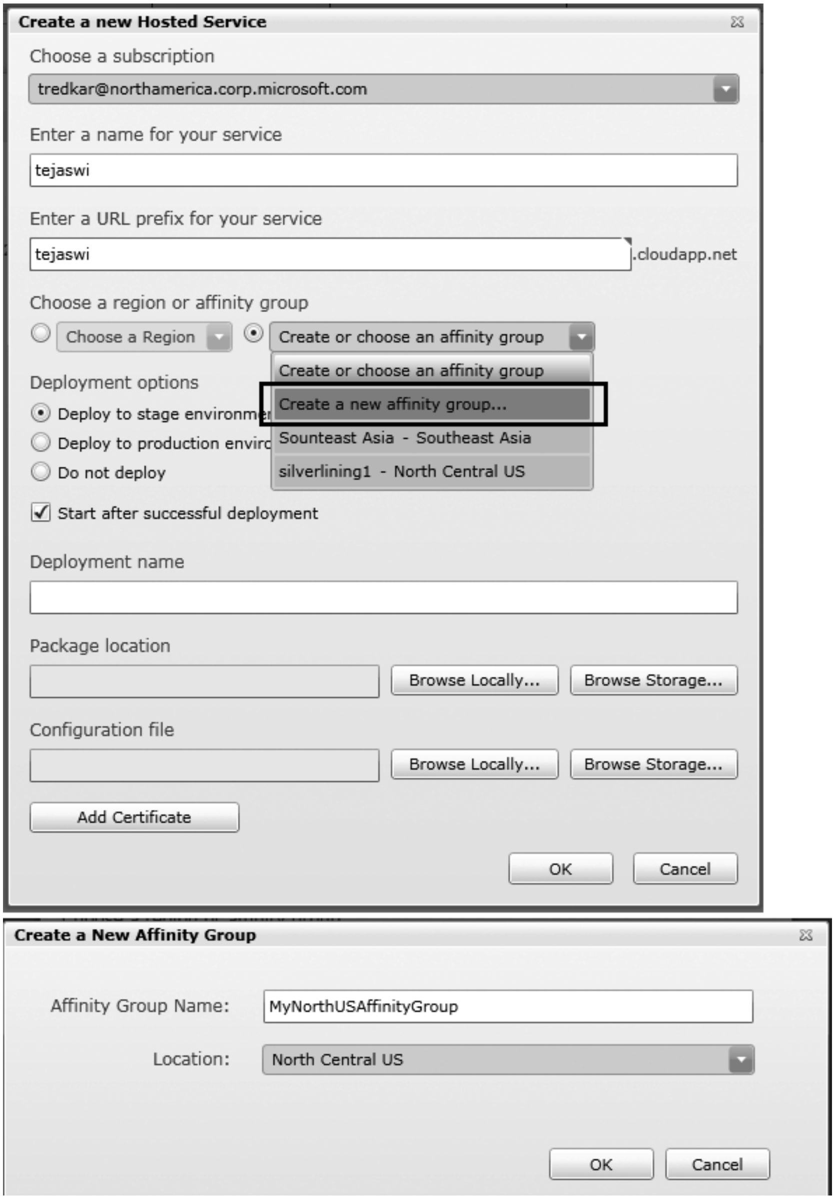
Geo-location support gives you the ability to choose the affinity of the Storage and Compute services to a particular geo-location.

Enabling Geographic Affinity

When you create a new storage account or a hosted services project, you can specify the location and affinity group for your project. The steps for creating a geographic affinity between a hosted service project and a storage account follow:

1. Log in to the Management portal and create a new Hosted Services project.
2. Give the project a name and a label. I have named my project tejaswi.
3. Select a hosted service URL.
4. Next, select Create or choose an Affinity group.

7. Select "Create a new affinity group" and choose a name and location for the affinity group, as shown in Figure 8-58.



Create a new Hosted Service

Choose a subscription: tredkar@northamerica.corp.microsoft.com

Enter a name for your service: tejaswi

Enter a URL prefix for your service: tejaswi.cloudapp.net

Choose a region or affinity group: ○ Choose a Region ● Create or choose an affinity group

Deployment options:
● Deploy to stage environme
○ Deploy to production enviro
○ Do not deploy

☑ Start after successful deployment

Deployment name

Package location

Configuration file

Create a New Affinity Group

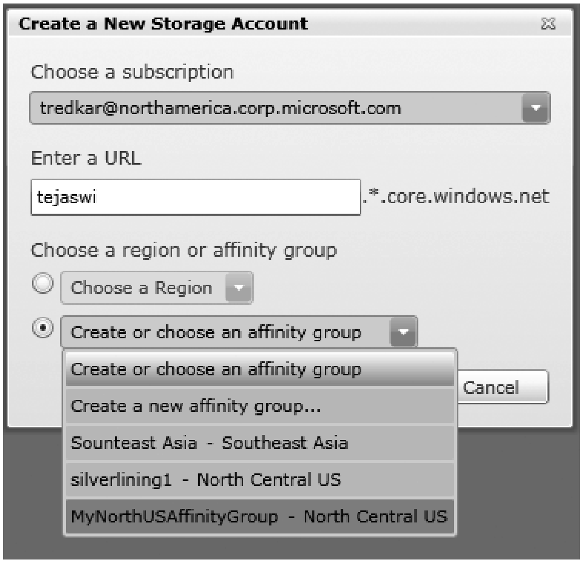
Affinity Group Name: MyNorthUSAffinityGroup

Location: North Central US

Create or choose an affinity group
Create a new affinity group...
Sounteast Asia - Southeast Asia
silverlining1 - North Central US
Browse Locally...
Browse Storage...
Browse Locally...
Browse Storage...
Add Certificate
OK
Cancel
OK
Cancel

Figure 8-58. *Hosted service affinity group*

8. Next, when you create a Storage service, you can associate it with the newly created affinity group as shown in Figure 8-59.



Create a New Storage Account

Choose a subscription

tredkar@northamerica.corp.microsoft.com

Enter a URL

tejaswi .*.core.windows.net

Choose a region or affinity group

○ Choose a Region

● Create or choose an affinity group

Create or choose an affinity group
Create a new affinity group...
Sounteast Asia - Southeast Asia
silverlining1 - North Central US
MyNorthUSAffinityGroup - North Central US
Cancel

Figure 8-59. *Affinity group*

Windows Azure Service Management

Unlike on-premise applications, where provisioning requires hardware and network infrastructure in place, the deployment of a cloud services requires only software provisioning. In a scalable environment where enterprises may need to provision multiple services across thousands of instances, you need more programmatic control over the provision process rather than configuring services using Windows Azure Management Portal or Visual Studio. Manually uploading service packages and then starting and stopping services from the portal interface works well for one or two services. For multiple large-scale services, it becomes a time-consuming and error-prone task. The Windows Azure Service Management API allows you to programmatically perform most of the provisioning functions via a REST-based interface. Using the Service Management API, you can script your provisioning and de-provisioning process end to end in an automated manner. In this section, I will cover some important functions from the Service Management API and also demonstrate some source code for you to build your own cloud service provisioning process.

Service Management API Structure

The Service Management API provides most of the functions you can perform on the storage services and hosted services from Windows Azure developer portal. The Service Management API categorizes the API operations into three primary sections: storage accounts, hosted services, and affinity groups. Operations on storage accounts mainly cover listing of accounts and generation of the access keys. Operation on hosted services cover listing of services, deploying services, removing services, swapping between staging and production, and upgrading services. The affinity groups operations are limited to listing and getting properties of affinity groups in your account.

■ **Note** You can find the Service Management API reference at `http://msdn.microsoft.com/en-us/library/ee460799.aspx`.

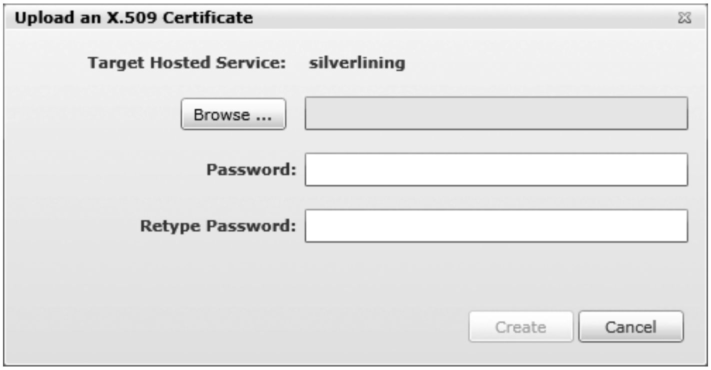
The Service Management API uses X.509 client certificates for authenticating calls between the client and the server.

Programming with the Service Management API

To start programming with the Service Management API, you must first create a valid X.509 certificate (or work with an existing one). You can use `makecert.exe` to create a self-signed certificate:

```
makecert -r -pe -a sha1 -n "CN=Windows Azure Authentication Certificate" -ss My -
len 2048 -sp "Microsoft Enhanced RSA and AES Cryptographic Provider" -sy 24
proazureservicemgmt.cer
```

Next, go to the services list in the Hosted Services section, select the Certificates folder under the service and click the Add Certificate button from the top menu. In the upload dialog box, select the certificate you want to upload and click OK.



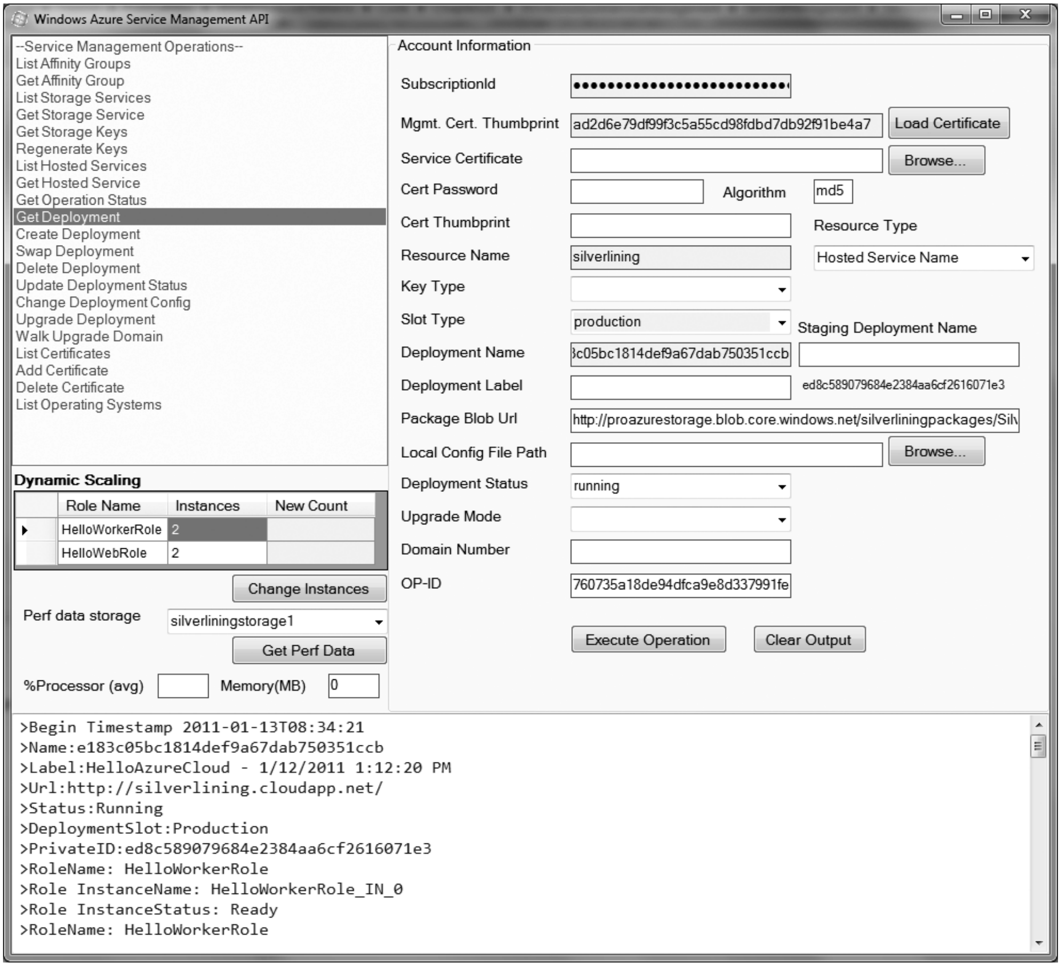
Upload an X.509 Certificate

Target Hosted Service: silverlining

***Figure 8-60.** Upload the API certificate*

Once the certificate is uploaded, you can call the Service Management REST API by passing the certificate as the `ClientCertificate` property of the `System.Net.HttpWebRequest` object, by using the `csmanage.exe` application from the Service Management API samples, or by building your own application. In Ch2Solution, I have created a sample Windows application that makes REST calls to the Service Management API. It uses the `Microsoft.Samples.WindowsAzure.ServiceManagement.dll` file from the service management code samples. The `csmanage.exe` uses the same assembly to make the API calls.

Eventually, the API assembly may become part of the Windows Azure SDK. Figure 8-61 illustrates the Service Management API Windows application in action.



	Role Name	Instances	New Count
▸	HelloWorkerRole	2	
	HelloWebRole	2	

Perf data storage: silverliningstorage1

%Processor (avg):

Memory(MB): 0

SubscriptionId: ••••••••••••••••••••••••••

Mgmt. Cert. Thumbprint: ad2d6e79df99f3c5a55cd98fdbd7db92f91be4a7

Service Certificate:

Cert Password:

Algorithm: md5

Cert Thumbprint:

Resource Type: Hosted Service Name

Resource Name: silverlining

Key Type:

Slot Type: production

Staging Deployment Name:

Deployment Name: 3c05bc1814def9a67dab750351ccb

Deployment Label:

ed8c589079684e2384aa6cf2616071e3

Package Blob Url: http://proazurestorage.blob.core.windows.net/silverliningpackages/Silv

Local Config File Path:

Deployment Status: running

Upgrade Mode:

Domain Number:

OP-ID: 760735a18de94dfca9e8d337991fe

```
>Begin Timestamp 2011-01-13T08:34:21
>Name:e183c05bc1814def9a67dab750351ccb
>Label:HelloAzureCloud - 1/12/2011 1:12:20 PM
>Url:http://silverlining.cloudapp.net/
>Status:Running
>DeploymentSlot:Production
>PrivateID:ed8c589079684e2384aa6cf2616071e3
>RoleName: HelloWorkerRole
>Role InstanceName: HelloWorkerRole_IN_0
>Role InstanceStatus: Ready
>RoleName: HelloWorkerRole
```

Figure 8-61. *The Service Management API Windows application*

The Service Management Operations section lists the operations that you can invoke on the Service Management API. The output textbox prints the output from the operations. The right-hand side of the user interface consists of input parameters. The input parameters textboxes turn yellow for required parameters. The input parameters are as follows:

- *Subscription Id*: You can get the subscriptionId from the Account page of the developer portal. This parameter is required by all the Service Management API operations.

- *Certificate Path*: This text box points to the API certificate file on the local machine. This certificate must match the one you uploaded to the portal.
- *Resource Type*: This drop-down lists the types of resource you want to access: Hosted Service, Storage Account, or Affinity Group.
- *Resource name*: You should type the name of the resource you want to access (e.g., storage account name, hosted service name, affinity group name).

The remaining input parameters are operation dependent. You can choose an operation from the Service Management operations list, enter input parameters and click Execute Operation. For example, to create a deployment in your hosted service account, you can do the following:

1. Select the Create Deployment operation.
2. Enter your Account SubscriptionId.
3. Select the API certificate from local machine (or add it to the app.config file for automatic loading).
4. Select Hosted Service Name as the Resource Type.
5. Enter the name of the Hosted Service you want to deploy your service to in the Resource Name text box.
6. Select the slot type (staging or production).
7. Choose a deployment name.
8. Choose a deployment label.
9. You have to then point to a service package (.cspkg) on a blob storage in the Package Blob URL text box.
10. Select the path to the ServiceConfiguration.cscfg file of the cloud service.
11. Click Execute Operation.

The OP-ID shows the operation ID returned by the method call, which you can use to track the operation status. To check the status of the deploy operation, select the Get Operation Status method, and click Execute Operation. The status gets displayed in the bottom window. Once the deployment is complete, you can run the deployment by selecting the Update Deployment Status method and selecting the “running” option from the deployment status drop-down. Similarly, you can execute other operations from the Service Management API.

■ **Tip** One of the more important uses of the Service Management API is in dynamic scaling of your Windows Azure Service. The Service Management API can change the configuration of your service on-the-fly, thus increasing or decreasing the number of instances of your roles. You track the performance counters from the Storage service for all the instances and then dynamically determine whether to scale-up or scale-down the service. You can find the latest source code for the Service Management application here: `azureplatformbook.codeplex.com`

Windows Azure Service Development Life Cycle

The objective of Windows Azure is to automate the service life cycle as much as possible. Windows Azure service development life cycle has five distinct phases and four different roles, as shown in Figure 8-62.

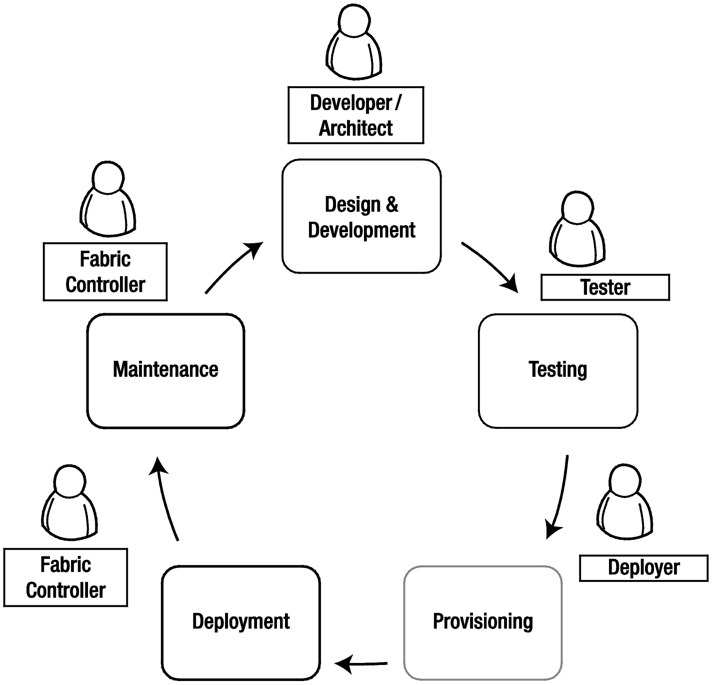


Figure 3-62. *The Windows Azure service development life cycle*

The five phases are as follows:

> *Design and development*: In this phase, the on-premise team plans, designs, and develops a cloud service for Windows Azure. The design includes quality attribute requirements for the service and the solution to fulfill them. This phase is conducted completely on-premise, unless there is some proof of concept (POC) involved. The key roles involved in this phase are on-premise stakeholders. For the sake of simplicity, I have combined these on-site design roles into a developer role.
>
> *Testing*: In this phase, the quality attributes of the cloud service are tested. This phase involves on-premise as well as Windows Azure cloud testing. The tester role is in charge of this phase and tests end-to-end quality attributes of the service deployed into cloud testing or staging environment.
>
> *Provisioning*: Once the application is tested, it can be provisioned to the Windows Azure cloud. The release engineer (deployer) role deploys the cloud service to the Windows Azure cloud. The deployer is in charge of service configurations and makes sure the service definition of the cloud service is achievable through production deployment in the Windows Azure cloud. The developer defines the configuration settings, but the deployer sets the production values. In this phase, the role responsibilities transition from on-premise to the Windows Azure cloud. The Fabric Controller in Windows Azure assigns the allocated resources as per the service model defined in the service definition. The load balancers and virtual IP address are reserved for the service.
>
> *Deployment*: In the deployment phase, the Fabric Controller commissions the allocated hardware nodes into the end state and deploys services on these nodes as defined in the service model and configuration. The Fabric Controller also has the capability of upgrading a service in running state without disruptions. The Fabric Controller abstracts the underlying hardware commissioning and deployment from the services. The hardware commissioning includes commissioning the hardware nodes, deploying operating system images on these nodes, and configuring switches, access routers, and load-balancers for the externally facing roles (e.g., the Web role).
>
> *Maintenance*: Windows Azure is designed with the assumption that failure will occur in hardware and software. Any service on a failed node is redeployed automatically and transparently, and the Fabric Controller automatically restarts any failed service roles. The Fabric Controller allocates new hardware in the event of a hardware failure. Thus, Fabric Controller always maintains the desired number of roles irrespective of any service, hardware, or operating system failures. The Fabric Controller also provides a range of dynamic management capabilities like adding capacity, reducing capacity, and service upgrades without any service disruptions.

In the previous sections, you learned how to design, develop, and deploy a cloud service to Windows Azure on production quality datacenter. In a traditional world, this would take several times longer. Be careful what you wish for, because empowerment also makes it easy to make bad decisions. In the next section, I will give you a checklist for following architecture best practices for developing Windows Azure applications.

Architectural Advice

Finally, here is a list of some practical advice that should serve you well going forward.

1. Performance- and load-test your application on Windows Azure to find out the optimum capacity needed. Don't decide on role sizes before testing.
2. Clearly separate the functionality of the Web role from the Worker role. Do not use Worker role to perform web functions by exposing HTTP (or HTTPS) endpoints.
3. Maintaining stateless role interfaces is important for load balancing and fault tolerance. Keep the roles stateless.
4. Use internal endpoints only for unreliable communications. For reliable communications, use Windows Azure queues.
5. User Worker roles effectively for batch and background processing.
6. Use Service Management API prudently for commissioning, decommissioning, and scaling of the role instances. Do not keep instances running idle for a long period of time, because you are using server resources and will be charged for it. Leverage third-party tools for scaling.
7. Do not use local storage for reliable storage; use Windows Azure Storage or SQL Azure as a reliable storage for storing data from roles.
8. Design the system for fault tolerance and always account for failure of role instances.
9. The Worker role works very well as Worker roles for high-scale compute.
10. For higher I/O operations, use large VM instances.
11. For large caching scenarios, use large VM instances, because they have larger memory capacity.
12. For distributed caching and session management, use Windows Azure AppFabric Caching.
13. Build dynamic scaling capabilities into all the Windows Azure services as a best practice. Most of the configuration for dynamic scaling is external and does not need changes to the source code of the roles.
14. Finally, do not deploy your cloud service for maximum capacity; deploy for optimum capacity, and dynamically provision more instances as demand increases, and vice versa.

Summary

In this chapter, we dove deeply into the computational features of Microsoft's Windows Azure cloud operating system. Through the examples, you were exposed to deploying Windows Azure Web role and Worker role instances, not only in the development fabric but also in the Windows Azure cloud. In the examples, you also learned how to access the configuration settings and local storage. Then, I briefly covered the geo-location and service management features of Windows Azure. In the examples in this chapter, we were storing and retrieving data from the local storage, which is local and machine dependent. The data will be lost as soon as the underlying machine is rebooted or the service redeployed. Windows Azure Storage provides you with persistent storage for storing highly available data that can be accessed from anywhere using REST-based API.

This final section on the Windows Azure platform provided you with a general overview of Windows Azure and a more in-depth look at the Windows Azure Compute architecture. These chapters should serve as a good starting point, and you can delve more deeply into topic with the complete *Windows Azure Platform*, Second Edition.

Bibliography

Apache Software Foundation. (n.d.). *Apache Hadoop*. Retrieved from `http://hadoop.apache.org`

Factor, A. (2001). *Analyzing Application Service Providers.* Prentice Hall.

Google. (n.d.). *Google AppEngine.* Retrieved from Google: `http://code.google.com/appengine`

Google. (n.d.). *Google Apps.* Retrieved from Google Apps: `http://www.google.com/apps/intl/en/business/index.html`

Mario Barbacci, M. H. (1995). *Quality Attributes.* Pittsburgh, Pennsylvania 15213: Software Engineering Institute, Carnegie Mellon University.

Microsoft Corporation. (n.d.). *About Windows Azure.* Retrieved from Windows Azure: `http://www.azure.com/`

Microsoft Corporation. (n.d.). *Windows Azure Pricing.* Retrieved from Windows Azure: `http://www.microsoft.com/azure/pricing.mspx`

Open ID Foundation. (n.d.). Retrieved from `http://openid.net/foundation/`

Staten, J. (2008). *Is Cloud Computing Ready For The Enterprise?* Forrester Research, Inc.