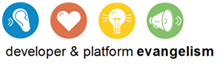
* 1. 

Demo Script

Hello Visual Studio 2010

* 1. Demo version: 1.1.0
  2. Last updated: 12/28/2010
  3. Prepared by: Jason Olson
  4. 

Contents

[Overview 3](#_Toc281309835)

[Key Messages 3](#_Toc281309836)

[Key Technologies 3](#_Toc281309837)

[Time Estimates 3](#_Toc281309838)

[Setup and Configuration 4](#_Toc281309839)

[Demo Flow 4](#_Toc281309840)

[Opening Statement 5](#_Toc281309841)

[Step-by-Step Walkthrough 5](#_Toc281309842)

[Launching Visual Studio 2010 6](#_Toc281309843)

[New Code Editing and Navigation Features 8](#_Toc281309844)

[One-Click Web Deployment 12](#_Toc281309845)

[Visual Studio as a Platform 13](#_Toc281309846)

[Summary 15](#_Toc281309847)

Overview

* 1. This document provides setup documentation, step-by-step instructions, and a written script for showing a demo of Visual Studio 2010. This document can also serve as a tutorial or walkthrough of the technology. Since Visual Studio 2010 is such a large project, we will not show all new features and improvements in this demo. Instead, you will examine the many new features and improvements coming to Visual Studio 2010 that make it a first-class IDE and powerful development platform. For additional demos of Visual Studio 2010, make sure to check out the Visual Studio 2010 and .NET Framework 4 Training Kit.

### Key Messages

In this demo you will see three key things:

* 1. First you will see how Visual Studio 2010 enables a powerful development experience through its many new features and improvements for developers.
  2. Second, you will see improvements coming to the way developers can configure and deploy web applications.
  3. Finally, you will see what new features are coming to Visual Studio 2010 that will enable the development of a large and vibrant third party ecosystem around the extension of Visual Studio.

### Key Technologies

This demo uses the following technologies:

* 1. Visual Studio 2010

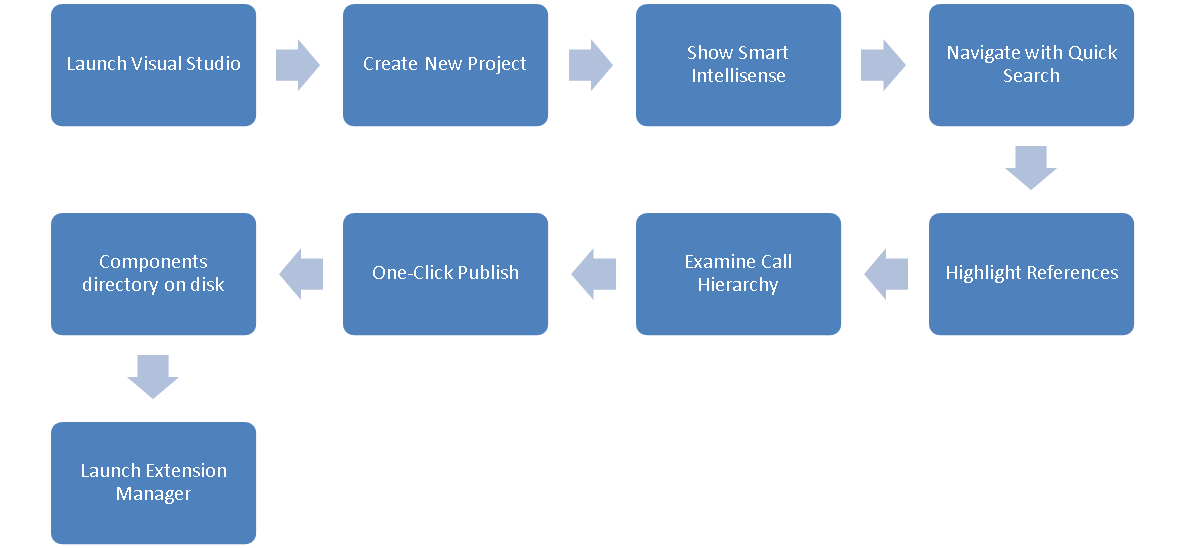
### Time Estimates

* + Estimated time to complete the demo: **10 min**.

Setup and Configuration

* 1. This demo does not have any configuration requirements other than a default install of Visual Studio 2010.

Demo Flow

* 1. The following diagram illustrates the high-level flow for this demo and the steps involved:
  2. 

Opening Statement

* 1. Visual Studio 2010 and .NET Framework 4 represent the next major release of Microsoft’s developer platform and toolset. In the next 10 minutes, we will build and deploy a simple web application using Visual Studio 2010. In the process, I want to show you many new and improved features that are coming to Visual Studio 2010 that enable a first-class development experience. With how large of a product Visual Studio 2010 is, we will obviously not show all improvements that are coming in this new release.

In this simple demo you will specifically see three key things:

* 1. First you will see how Visual Studio 2010 enables a powerful development experience through its many new features and improvements for developers.
  2. Second you will see how improvements coming to the way developers can configure and deploy web applications.
  3. Finally, you will see what new features are coming to Visual Studio 2010 that will enable the development of a large and vibrant third party ecosystem around the extension of Visual Studio.

Step-by-Step Walkthrough

1. This demo is composed of the following segments:
   * Launching Visual Studio 2010
   * New code editing and navigation features
   * One-Click web deployment
   * Visual Studio’s platform improvements

### Launching Visual Studio 2010

|  |  |  |
| --- | --- | --- |
| Action | Script | Screenshot |
| * 1. Start Visual Studio 2010. | * Let’s launch Visual Studio 2010 * One of the first things you may or may not notice is that the VS shell is now written in WPF. This is a very powerful decision and brings forth a lot of potential for very cool features that could not be implemented previously. * You will also notice a new start page. Once again, this new start page leverages WPF and, hence, is XAML itself. This makes it easily customizable so you can customize it to your own specific needs (perhaps a custom team start page with links to internal coding practices, build servers, etc.) |  |
| * 1. Highlight project(s) in Recent Projects list. | * One of the features afforded to us by the adoption of WPF is easier management of our **Recent Projects** list for instance. * Not only can we easily remove projects that we do not wish to be on the list anymore, but I can also pin any projects that I may work on a lot or view as important and wish to keep on the top of my list. |  |
| * 1. Create a new project by clicking **New Project** on the start page.   2. Expand .NET Framework version drop-down. | * First, let’s go ahead and create a new web project. * One feature we want to draw your attention to is multi-targeting. Similar to previous releases of Visual Studio, you have the choice of targeting different versions of the .NET Framework when developing projects in Visual Studio 2010. * Multi-targeting allows us to use the latest and greatest tools at our disposal, like Visual Studio 2010, while continuing to be able to develop our current applications against their currently targeted version of the framework. |  |
| * 1. Create web project by choose Visual Basic or C# -> Web -> ASP.NET Empty Web Application. | * We are going to go ahead and create a new ASP.NET Empty Web Application. * Let’s just name the project “HelloWeb” for right now. |  |

### New Code Editing and Navigation Features

|  |  |  |
| --- | --- | --- |
| Action | Script | Screenshot |
| * 1. Right Click in the Project-> Add New Item.   2. Write Web Form in the search box and name it Page Default.aspx.   3. Open Default.aspx, and choose the Design surface if not chosen already.   4. Open **Toolbox**, expand “**Standard**”.   5. Drop a **Button** and **Label** onto our page from the **Toolbox**.   6. Rename **Button** -> btnSend. Change **Button.Text** -> “Press Me”.   7. Rename **Label -> lblMessage**. Remove value from Label.Text. | * So you’ve seen some simple features that are coming that improve Visual Studio’s launch-time experience, let’s take a look at some of the new features that will make a developer’s life easier with what we do most: deal with code. * Another useful feature is the ability to search when inserting a new item. * For the sake of this demo, we will just create a simple Button and Label on a clean default aspx page. |  |
| * 1. **Double-Click** our **Button** in the designer to create a click event handler in code-behind.   2. In the event handler, type “Mes”.   3. Choose lblMessage from Intellisense and set its Text property to “Clicked!” | * We are going to go ahead and create an event handler for our Button’s click event. * In the event handler, we are going to simply change the text in our Label. * You will notice that even though I typed “Mes”, our Label is still found in IntelliSense even though it does not start with “Mes.” This is because in Visual Studio 2010, Intellisense is much smarter and will search for “sub-string” occurrences for the text as well, rather than just “begins with” occurrences. * Let’s go ahead and change the text to “Clicked!” |  |
| * 1. Highlight the line of code we just wrote.   2. Extract Method: Right Click -> Refactor -> Extract Method.      1. **Note:** The **Refactor** Menu is not available in Visual Basic.   3. Call the method ChangeText.   4. Add a string parameter named **message** to the ChangeText method, and pass in “Clicked!” from the Button’s click event handler. | * We are going to go ahead and use this code again, so we will just refactor this code into a common method called ChangeText. |  |
| * 1. **Press “ctrl+,”** (hold down control and press comma) to bring up **Quick Search** dialog.   2. Enter “Page” as search item.   3. Delete “Page” search item and type “PL” instead.   4. Press enter to “OK” and accept dialog. | * Another challenge for developers is to be able to navigate around a large code base easily. This becomes much easier in Visual Studio 2010 with the introduction of Quick Search. * In this case, let’s use it to navigate to our Page\_Load method (though this feature is really useful when searching cross-project or cross-file within a solution). * As you can see, when we enter “Page”, our method is found as expected. * What is really cool though, is that Quick Search uses a smart search algorithm that isn’t just substring search. It is fully aware of Pascal Casing for instance. * Let’s search for “PL” (for **P**age\_**L**oad) instead. As you can see, our method is still found. When searching for other classes in your project, this becomes a very powerful feature to use for navigation around your codebase. * Let’s go ahead and accept in order to open our Page\_Load method. |  |
| * 1. In the **Page\_Load** method, call our **ChangeText** to set text to *“Not Clicked”.* | * To add a new default value, we will simply call our new ChangeText method with the value “Not Clicked”. |  |
| * 1. Click the cursor on the name of our **ChangeText** method. | * Easy visualization of where all references to a method or variable are within a single code file is possible now with the introduction of Highlight References in the editor. * We are just going to click the ChangeText method, and you can see all references to that method are automatically highlighted for us. |  |
| * 1. Right Click on the name of the ChangeText method -> View Call Hierarchy. | * Another feature that makes it easy to really dig into the relationships between various methods and lines of code is the new Call Hierarchy feature. * The problem is that trying to understand external and internal relationships of a method today requires a lot of digging around the results from “Find All References.” * Now, by right-clicking on our ChangeText method, we can select “View Call Hierarchy” from the context menu. * As you can see, we can easily dive into all the various calls that are made to ChangeText as well as all the calls that are made from ChangeText. * By selecting one of the calls either to or from ChangeText, we can also see the specific line of code where the call is made. |  |

### One-Click Web Deployment

|  |  |  |
| --- | --- | --- |
| Action | Script | Screenshot |
| * 1. In the **Publish bar**, pull down the drop down and click “**<New...>**”. | * For web developers, a fairly common task is deploying your website to a specific environment (QA, Staging, Live, etc.). * Unfortunately, robust deployment tools like MS Deploy are only available outside of Visual Studio today. * With Visual Studio 2010, MSDeploy is integrated directly into Visual Studio. Once you have your profiles configured, you can easily deploy to a given environment with a single click. * This isn’t limited to just MSDeploy either. It also supports other modes of deployment, like via FTP. * Let’s go ahead and create a new profile for our QA environment. |  |
| * 1. Name profile **QA** by typing “QA” into the **Publish Profile** drop down.   2. Add **http://qa.contoso.com/HelloWeb** to the **Site/Application**.   3. Add **http://qa-web:8172/MsDeploy.axd** to the **Service Url**.   4. Click **Save** button, and then click **Close**.   5. Choose **QA** from **Publish bar** drop down, and hover over “**Publish Web**” button. | * You can see that not only can we specify the method of publishing, we can also specify the site/application, and even mark it as an IIS application on the destination. * Let’s add some information on our environment to our profile. * We are going to go ahead and save this. * We can know have our QA profile chosen and with one click easily deploy to the designated profile. * Combining this tool with other new features in Visual Studio 2010 (like web.config transformations) makes for a much more powerful web publishing experience than before. |  |

### Visual Studio as a Platform

|  |  |  |
| --- | --- | --- |
| Action | Script | Screenshot |
| * 1. Open Windows Explorer   2. Navigate to “C:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE\CommonExtensions” or C:\Program Files (x86)\Microsoft Visual Studio 10.0\Common7\IDE\CommonExtensions”. | * A lot of work is going into making Visual Studio a first-class platform, which third parties can “embrace and extend.” * Part of this effort is making it easier for developers to extend Visual Studio. * With Visual Studio 2010, creating an editor extension for example is as easy as compiling an assembly and simply dropping it into a specific directory in the Visual Studio installation. * Next time Visual Studio 2010 is started, it will automatically pick up the new extension. * While this is simple for developers, it is not the best experience for end-users. |  |
| * 1. Re-Open Visual Studio.   2. Open **Tools -> Extension Manager**.   3. Browse the **Online Gallery** and select an extension not currently installed.   4. Show the **Download** button. | * To help make the 3rd party ecosystem of Visual Studio much easier to digest, there is a new Extension Manager built into Visual Studio that automatically plugs into the online VS Gallery. * Using Extension Manager, a Visual Studio user can easily find all sorts of extensions that have been written for Visual Studio, and easily install or uninstall them directly from the Extension Manager itself. * Browsing, searching, downloading and installing an extension is very simple and straightforward. * This gives users the power to customize Visual Studio to best suit their own development needs. * The extensions are installed under \Users\{UserName}\AppData\Local\Microsoft\VisualStudio\10.0\Extensions. |  |

Summary

* 1. In this demo, you saw how developers have a lot more tools at their disposal in which to investigate and understand the code of an application. You also got a quick look at how much more powerful Visual Studio 2010 is for web developers needing to configure, manage, and deploy-to many different environments. Finally, we looked at the new platform capabilities like Extension Manager and the new Editor that will enable Visual Studio to foster the development of a large and vibrant third party ecosystem around the extension of Visual Studio itself.

Whether you are a web developer or client developer, managed developer or native developer, there truly is not a single developer that is not affected by this new release of Microsoft’s developer platform and toolset. In later demos, we will see what features and enhancements are coming to the .NET Framework that will make developers more efficient and capable than they were with previous releases.