Hands-On Lab
Lab Manual

In this lab you will develop a Windows Forms application and learn how to take advantage of .NET Framework’s No-Touch Deployment technology as well as the .NET Application Updater Component.
Please do not remove this manual from the lab
The lab manual will be available from CommNet
Contents

LAB 1: CREATE AND DEPLOY THE SAMPLE APP ........................................................................................................ 1
Lab Objective .......................................................................................................................................................... 1
Exercise 1 – Develop the test application ........................................................................................................ 1
  Task 1 – Create a Windows Forms application ............................................................................................ 1
  Task 2 – Add basic functionality .................................................................................................................. 2
Exercise 2 – Prepare for No-Touch Deployment ............................................................................................. 3
  Task 1 – Create a virtual directory in IIS for deployment ........................................................................ 3
Exercise 3 – Using No-Touch Deployment .................................................................................................... 4
  Task 1 – Build and test your application .................................................................................................... 4
  Task 2 – Create a new version of your application and test deployment ................................................ 4
Lab Summary .................................................................................................................................................... 5

LAB 2: A MORE COMPLEX DEPLOYMENT STRATEGY .......................................................................................... 6
Lab Objective ...................................................................................................................................................... 6
Exercise 1 – Configure your application for use with the .Net Application Update Manager ..................... 6
  Task 1 – Configure IIS for use with .Net Application Update Manager .................................................. 6
  Task 2 – Configure your application to use the .Net App Update Manager ........................................... 7
Exercise 2 – Deploy and test the application ................................................................................................. 9
  Task 1 – Prepare the application for deployment ....................................................................................... 9
  Task 2 – Create a new version and deploy it ............................................................................................... 10
Lab Summary ................................................................................................................................................... 11
Lab 1: Create and deploy the Sample App

Lab Objective

Estimated time: 20 minutes

The objective of this lab is to develop the application we will use for No-Touch Deployment and for the .NET Application Updater component. In the second exercise we will prepare this application and the environment for deployment with No-Touch Deployment. Finally, we will test this deployment strategy by creating a new version of our application and rolling it out to verify that the update functions as expected.

- Develop the test application
- Prepare for deployment
- Using No-Touch Deployment

Exercise 1 – Develop the test application

In this exercise, you will create a simple Windows Forms application in the .NET Framework-supported language of your choice. Samples are in Visual Basic .NET and C# so pick the one with which you are most comfortable.

Task 1 – Create a Windows Forms application

- Select Start | All Programs | Microsoft Visual Studio .NET | Microsoft Visual Studio .NET
- Click New Project.
- Depending on what language you want to use, select Visual Basic Projects or Visual C# Projects in the Project Types pane.
- Select Windows Application in the Templates pane.
- Change the Name: to DeploymentTest.
- Click OK.

The DeploymentTest project is added to the solution and the form designer for Form1 now appears in the development environment.
Task 2 – Add basic functionality

- Drag a label control onto Form1 from the Toolbox on the left.
- Erase “label1” from the Text property in the Properties pane on the right.
- In the properties pane on the right change the Font property to a larger size font for label1. You can also change the type face if you would like. (See Figure 1.1)
- Double Click the body of Form1.
- Scroll to the top of Form1.cs in the code view.
- Add the following line of code below the line “using System.Data” for C# 
  ```csharp
  using System.Reflection;
  ```
  and the following below the line “Imports System.Data” for VB.NET
Imports System.Reflection

**Note:** This will allow you to use the Assembly class to display your application’s version number.

- Scroll back down to the Form1_Load function and add the following code for C#
  ```csharp
  label1.Text = Assembly.GetExecutingAssembly().GetName().Version.ToString();
  ``

  And the following for VB.Net
  ```vbnet
  label1.Text = [Assembly].GetExecutingAssembly().GetName().Version.ToString()
  ``

  **Note:** This will set your label’s text to the Version number and change the background color of the label to red.

- Press F5 in order to start your application
- When your application starts up you will see the version number and the red background in label1.

### Exercise 2 – Prepare for No-Touch Deployment

In this exercise, you will do all of the necessary set up work to prepare the application you just created for No-Touch Deployment.

#### Task 1 – Create a Virtual directory in IIS for deployment

- Open IIS Manager: Select Start | All Programs | Administrative Tools | Internet Information Services (IIS) Manager.
- Expand Local Computer | Web Sites
- Right click on Default Web Site
- Point to New, click on Virtual Directory…
- Click Next.
- Change the Alias textbox to “NoTouchDeploy”.
- Click Next.
- Click Browse.
- Expand Local Disk (C:) | Inetpub | wwwroot.
- Click Make New Folder.
- Change the name of the new folder to “NoTouchDeploy”.
- Click OK.
• Click **Next** on the **Virtual Directory Access Permissions** page, the defaults will work for our application.

• Click **Finish**.

• Close **Internet Information Services (IIS) Manager** using the X in the top right corner.

• Switch to Visual Studio to finish set up.

• In the solution explorer, right click the DeploymentTest project and select **Properties**.

• Select **Configuration** in the left hand pane

• Change the **Output Path** to point to the virtual directory you just added to IIS, "C:\Inetpub\wwwroot\NoTouchDeployment"

• Click **OK**.

### Exercise 3 – Using No-Touch Deployment

In this exercise, you will deploy and test your application using No-Touch Deployment.

#### Task 1 – Build and test your application

• Click **Build | Build DeploymentTest**  
  **Note:** This will rebuild your project into the IIS virtual directory you created earlier.

• Click **Start | All Programs | Internet Explorer**.

• In the address window of Internet Explorer type :
  "http://localhost/NoTouchDeployment/DeploymentTest.exe"

• When your application starts verify that this is indeed version 1, the red label version of your application.

• Close your application by clicking the X in the top right corner of the form.

#### Task 2 – Create a new version of your application and test deployment

• Go back to Visual Studio.Net.

• In the **Solution Explorer** pane on the right double click on **Form1.cs** or **Form1.vb** depending on the language you chose.

• Scroll down to the form load event handler, **Form1_Load(...)**

• Change the color you are setting the label background from Red to your favorite color (if red is your favorite use your second favorite).

  **Note:** You can pick any color here besides Red. When you delete Red also delete the period preceding it so that when you hit the period again a drop down list of all of the available colors will appear.

• Click **Build | Build DeploymentTest**.
• Go back to Internet Explorer by clicking on Internet Explorer in the taskbar.
• Reload your application by pressing F5 or retyping the address above.
• When your application starts verify that this is indeed the new version of your application with your chosen color as the label’s background.

   **Note:** You will notice too that the number displayed on your label has incremented. By default Visual Studio.Net increments the version of an application every time it is built.

**Lab Summary**

In this lab you performed the following exercises:

- Develop the test application
- Prepare for No-Touch Deployment
- Using No-Touch Deployment

In this lab, you created an application which can be deployed with No-Touch Deployment, the Windows application deployment mechanism built into the .NET Framework. The second exercise accomplished all of the setup tasks necessary to actually deploy and run your application using No-Touch Deployment. With the application built and the set up complete the third lab walked you through building and testing the deployment. At this point you should be familiar with how you would use this basic deployment strategy for any Windows Forms application.
Lab 2: A more complex deployment strategy

Lab Objective

Estimated time to complete this lab: 30 minutes

The objective of this lab is to learn how to deploy an application using the .NET Application Updater component. This component was originally available on www.windowsforms.net (http://windowsforms.net/articles/appupdater.aspx). It has since been updated with some new features and will soon be released and supported by Microsoft directly. We will be using a beta version of this component which will be complete and available on MSDN (http://msdn.microsoft.com/patterns) shortly after TechEd 2003. This component manages checking for updated versions of an application on a remote Web server, downloading any updates and then configuring the local install to run the new version. Applications using this component can automatically run offline and can be rolled back to use the previous version should the update prove unsatisfactory in any way. The next major release of the .NET Framework will natively feature all of this functionality, but this component provides an excellent interim solution.

- Configure your application for use with the .NET Application Updater component
- Deploy and test your application

Exercise 1 – Configure your application for use with the .NET Application Update Manager

In this exercise you will configure your Windows Forms application for use with the .NET Application Updater component.

Task 1 – Configure IIS for use with .Net Application Update Manager

- Click Start | All Programs | Administrative Tools | Internet Information Services (IIS) Manager.
- Expand Local Computer | Web Sites
- Right click on Default Web Site
- Point to New, click on Virtual Directory…
- Click Next.
- Change the Alias textbox to “AppUpdater”.
- Click Next.
- Click Browse.
- Expand Local Disk (C:), expand Inetpub, click on wwwroot.
- Click Make New Folder.
- Change the name of the new folder to “AppUpdater”.
- Click OK.
Click **Next** on the *Virtual Directory Access Permissions* page, in addition to the two **Read** access permissions check **Browse** so that the updater can browse the directory for new versions.

Click **Finish**.

Close **Internet Information Services (IIS) Manager** using the X in the top right corner.

Copy serverconfig.xml from \Microsoft Hands-on-Lab\Dev-HOL12\HOL_AppUpdater\ to \Inetpub\wwwroot\AppUpdater.  
**Note**: This file contains all of the information needed to make a new version of your application available. It is a simple XML file so feel free to open it up and see what it contains. Also, we will be updating it later in order to publish a new version of our application.

**Task 2 – Configure your application to use the .NET Application Updater component**

Switch to Visual Studio and open the DeploymentTest project if it isn’t already open.

In the Solution Explorer right click **References | Add Reference**.

Click **Browse** in the top right corner of the **Add Reference** dialog.

Navigate to \Microsoft Hands-on-Lab\Dev-HOL12\HOL_AppUpdater\ and select Microsoft.ApplicationBlocks.ApplicationUpdater.dll,
Microsoft.ApplicationBlocks.ExceptionManagement.dll,
Microsoft.ApplicationBlocks.ExceptionManagement.Interfaces.dll, and
Microsoft.ApplicationBlocks.ApplicationUpdater.Interfaces.dll

Click **Open** and then click **OK**

Right click your project in Solution Explorer and select **Add | Add Existing Item…**

Navigate to \HOL_AppUpdater, select App.config, and click **Open**

**Note**: App.config contains all of the information your application needs to update itself including the location of the update server (**http://localhost** in this example), the location for temporary download files, and where to install the latest version of your application after the download is complete.

At the top of Form1.cs or Form1.vb add the following for **C#**:

```csharp
using Microsoft.ApplicationBlocks.ApplicationUpdater;
```

or the following for **VB.Net**

```vbnet
Imports Microsoft.ApplicationBlocks.ApplicationUpdater
```

At the top of class **Form1**, add the following code before the constructor for form1.

```csharp
ApplicationUpdateManager appUpdater = null;
```

Or for **VB**

```vbnet
Private appUpdater As ApplicationUpdateManager = Nothing
```

Scroll down to the **Form1_Load** function and add the following code to the existing code for **C#**
if (appUpdater == null)
{
    appUpdater = new ApplicationUpdateManager();
    appUpdater.UpdateComplete+=new ApplicationUpdateManager.UpdateCompleteHandler(OnComplete);
    appUpdater.StartUpdater();
}

Or the following for VB.Net

If appUpdater Is Nothing Then
    appUpdater = New ApplicationUpdateManager
    AddHandler appUpdater.UpdateComplete, AddressOf OnComplete
    appUpdater.StartUpdater()
End If

• Now we need to add the Update Complete event handler. This event will be raised when your application has successfully been updated and when you must restart to see the affects. This process is different for VB.Net and C# so choose the appropriate set of steps.

For C#
• Switch to the design view of Form1 by clicking on Form1.cs[Design]
• Right click on Form1 and choose Properties.
• Click on the lightning bolt icon.
• Scroll to the Closing event, type “OnForm1Closing” and press enter
  Note: This will automatically create the OnForm1Closing event handler and open the code view for that method.
• Add the following code to OnForm1Closing(…)
  if (appUpdater != null)
      appUpdater.StopUpdater();
  Note: This will ensure that the Update Manager is stopped when our application closes.
• Add this code above the Form1_Load function. This function will be called when the application updated event has been received.
  private void OnComplete(object sender, EventArgs e)
  {
      MessageBox.Show("Update Complete!");
  }
• Build your application by selecting the Build | Build DeploymentTest menu command.

For VB.Net
• Stay in the code view of Form1.vb.
• In the Left drop down at the top of the Form1 code editor select (Form1 events).
• Now, in the Right hand drop down select the Closing event.
Note: This will automatically create the Form1_Closing method and open the code view for that method.

- Insert the following code in the Form1_Closing method. This will ensure that the Update Manager is stopped when our application closes.
  
  If Not (appUpdater Is Nothing) Then
    appUpdater.StopUpdater()
  End If

- Add this code for VB.Net above the Form1_Load function. This function will be called to let us know the application updated event has been received.
  
  Private Sub OnComplete(ByVal sender As Object, ByVal e As EventArgs)
    MessageBox.Show("Update Complete!")
  End Sub

Exercise 2 – Deploy and test the application

In this exercise you will deploy your Windows Forms application with the .NET Application Updater component and test to verify that it was in fact updated.

Task 1 – Prepare the application for deployment

- Open Windows Explorer by clicking Start | Windows Explorer.
- Navigate to C:\Program Files\.
- Create a home directory for your application by selecting the File | New | Folder menu command.
- Change the folder name to "DeploymentTest"
- Create a folder for version 2.0.0.0 of your application by selecting the File | New | Folder menu command.
  
  Note: We have already built version 1.0.0.0 of this application for No-Touch Deployment so we’ll use 2.0.0.0 as our base application for this section.

- Change the folder name to “2.0.0.0”
- Now, copy AppStart.exe and AppStart.exe.config from C:\HOL_AppUpdater to C:\Program Files\DeploymentTest.
  
  Note: AppStart.exe starts up the latest version of your application for you depending on the version information stored in AppStart.exe.config. In this case that is version 2.0.0.0. When your application is updated by the .NET Application Updater component, AppStart.exe.config will be updated with the latest version information so that when you run your program the new version will be loaded.

- Switch back to Visual Studio .NET and in the solution explorer, right click the DeploymentTest project and select Properties.
- Select Configuration in the left hand pane
- Change the Output Path to the folder you just created.
  
  “C:\Program Files\DeploymentTest\2.0.0.0”
• Click OK.
• Build your application by clicking Build | Build DeploymentTest.
• To test to make sure you have configured everything correctly, double click AppStart.exe. When the application starts the top label should be your favorite color from Exercise 3.Task 2 above to signify version 2.

Task 2 – Create a new version and deploy it.

• Create version 3.0.0.0 of your application by changing the background color of label1 on Form1. In Form1_Load, change the background to green instead of blue. Do this by changing this line:
  to this:
• Now we must create the Web site from which the application will be updated. In this case we will be creating a virtual directory which has the new version, 3.0.0.0 as well as an XML file with the update information. Navigate to C:\Inetpub\wwwroot\AppUpdater using Windows Explorer.
  • In this folder create a new folder for the new version, 3.0.0.0. Click File | New | Folder.
• Change the folder name to “3.0.0.0”.
• Switch back to Visual Studio in order to build and deploy your solution.
• In the solution explorer, right click the DeploymentTest project and select Properties.
  • Select Configuration in the left hand pane
  • Change the Output Path to the virtual folder you just created for IIS. “C:\Inetpub\wwwroot\AppUpdater\3.0.0.0”
• Build version 3.0.0.0 of your application by selecting Build | Build DeploymentTest.
• When your application runs it will first check http://localhost/AppUpdater/serverconfig.xml to see if a new version is available. Up until now only version 2.0.0.0 has been available and that’s the same version we have been running. In order to tell the updater that a new version is available we must change serverconfig.xml on the local Web server. Navigate to C:\Inetpub\wwwroot\AppUpdater\ using Windows Explorer.
  • Right click serverconfig.xml and select Edit with Visual Studio .NET 2003. This will open serverconfig.xml in Visual Studio and allow you to make the necessary changes.
  • Change the value of the availableVersion node from 2.0.0.0 to 3.0.0.0.
  • Change the value of the updateLocation node from http://localhost/AppUpdater/2.0.0.0 to
  http://localhost/AppUpdater/3.0.0.0.
• Save this file using the File | Save ServerConfig.xml command.
• Now, go back to the currently installed version of your application and start it with AppStart.exe in C:\Program Files\DeploymentTest.

• When the app starts it will automatically check whether it is running the most current version. In this case it isn’t so it will begin to download the latest version. You will see the temporary files first in C:\Program Files\DeploymentTest\newfiles\. When the download and update is complete a messagebox will pop up to notify you that it is done. After you receive this messagebox close the application and restart it with AppStart.exe. You will immediately see that this is the new version of your application signified by the color in the label.

Lab Summary

In this lab you performed the following exercises.

- Configure your application for use with the .NET Application Updater component
- Deploy and test your application

In this lab, you configured your existing DeploymentTest application and IIS for use with the .Net Application Update Manager. Once it was configured you tested and then deployed a new version. When you started your app the second time it automatically downloaded the new version and informed you that a new version was available and ready. After these exercises you should be familiar with how to set up an application for deployment with the .NET Application Updater component. There are many more helpful tasks which can be accomplished with this component. Be sure to check out the released version on MSDN shortly.