



COMPSCI 280 S1 2011 Enterprise Software Development

Developing Windows Applications with Visual Studio 2010
Deployment (Extra Reference Only)



Agenda & Reading

- ▶ **Agenda:**
 - ▶ ClickOnce Deployment
 - ▶ Windows Install Deployment
- ▶ **Recommended Reading:**
 - ▶ ClickOnce Deployment
 - ▶ [http://msdn2.microsoft.com/en-us/library/t71a733d\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/t71a733d(VS.80).aspx)
 - ▶ Windows Installer Deployment
 - ▶ [http://msdn2.microsoft.com/en-us/library/2kt85ked\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/2kt85ked(VS.80).aspx)
- ▶ **Hands-On Lab:**
 - ▶ Lecture 16 Lab



Introduction

- ▶ **Deployment is the process by which you distribute a finished application or component to be installed on other computers.**
- ▶ **In Visual Studio 2008, you can deploy applications using either**
 - ▶ ClickOnce deployment, or
 - ▶ With a traditional Setup using Windows Installer deployment technology
- ▶ **There are many factors to consider when choosing a deployment strategy:**
 - ▶ the type of application,
 - ▶ the type and location of users,
 - ▶ the frequency of application updates, and
 - ▶ the installation requirements.



ClickOnce Deployment

- ▶ **ClickOnce Deployment**
 - ▶ It allows you to publish Windows-based applications to a centralized location (a Web page, a network file share, or media such as a CD-ROM) for simplified installation.
 - ▶ The user installs or runs the application from that location in a single step.
- ▶ **Features:**
 - ▶ It greatly simplifies the process of installing and updating an application.
 - ▶ But you do not get the power of Windows Installer deployment, which provides you with greater flexibility.
 - ▶ ClickOnce deployed applications are self-updating and are the best choice for applications that require frequent changes.
- ▶ **The core ClickOnce deployment architecture is based on two XML manifest files:**
 - ▶ an application manifest, and
 - ▶ a deployment manifest.



Manifest Files

- ▶ **Two manifest files:**
 - ▶ The application manifest
 - ▶ describes the application itself, including the assemblies, the dependencies and files that make up the application, the required permissions, and the location where updates will be available.
 - ▶ The deployment manifest
 - ▶ describes how the application is deployed, including the location of the application manifest, and the version of the application that clients should run.
- ▶ **How ClickOnce Deployment Works**
 - ▶ The deployment, application manifest and all of the application files are copied to a deployment location
 - ▶ The end users can download and install the application by clicking an icon representing the deployment manifest file on a Web page or in a folder.
 - ▶ The end user is presented with a simple dialog box asking the user to confirm installation, after which installation proceeds and the application is launched



Application

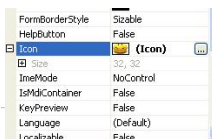


- ▶ **Installing**
 - ▶ The application is added to the user's Start menu and to the Add/Remove Programs group in the Control Panel.
 - ▶ Unlike other deployment technologies, nothing is added to the Program Files folder, the registry, or the desktop, and no administrative rights are required for installation
- ▶ **Versioning**
 - ▶ ClickOnce Publish properties are used to specify when and how often the application should check for updates.
 - ▶ Publish properties can be employed to make updates mandatory or to roll back to an earlier version.
 - ▶ When the application developer creates an updated version of the application, he or she also generates a new application manifest and copies files to a deployment location—usually a sibling folder to the original application deployment folder.



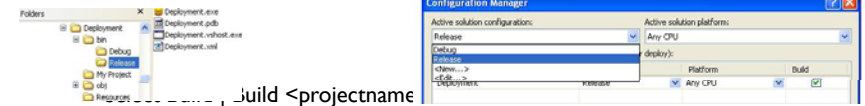
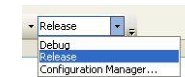
Before we start ...

- ▶ **Advanced Setting**
 - ▶ **Application Icon**
 - ▶ The Icon property for a project specifies the icon file (.ico) that will be displayed for the compiled application in Windows Explorer and in the Windows taskbar.
 - ▶ To specify an Application Icon
 - In Solution Explorer, right-click the application project and choose Properties.
 - Click the Application tab
 - Select an icon file from the Icon drop-down list
 - (demo: I6.1_01AddIconAndRelease.wmv)
 - ▶ **A Form Icon**
 - ▶ A form's icon designates the picture that represents the form in the taskbar as well as the icon that is displayed for the control box of the form
 - ▶ To specify a Form Icon
 - In Solution Explorer, right-click the form and choose Properties
 - Select an icon file from the Form.Icon Property



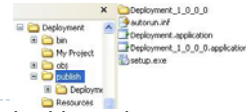
Compiling a Release Version

- ▶ **Compiling a Release build**
 - ▶ Build the Release version for the final release distribution
 - ▶ The Release configuration of your program is fully optimized and contains no symbolic debug information
 - ▶ To build a release version
 - ▶ On the Standard toolbar, click Release from the Solution Configurations list box, or
 - Choose Build | Configuration Manager from the menu
 - Select Release from the Active solution configuration dropdown list. Click Close.





Publish Wizard



- ▶ You can publish the application using the Publish Wizard; additional properties related to publishing are available on the Publish page of the Project Designer.
- ▶ To publish to a CD-ROM or DVD-ROM
 - ▶ In Solution Explorer, right-click the application project and choose Properties.
 - ▶ Click the Publish tab and click the Publish Wizard button.
 - ▶ Choose Publishing Location
 - ▶ Select "From a CD-ROM or DVD-ROM"
 - ▶ Select "The application will not check for updates"
 - ▶ Click Finish
 - ▶ Optional: Select Options from Install Mode and Settings
 - ▶ Publish name (the name of the folder that contains the application's icon in the Start menu. If it is empty, the value of the computer's RegisteredOrganization property will be used)
 - ▶ Product name (the name of the entry in the Start menu, and the name that shows in Add or Remove Programs. If it is empty, the assembly name is used.)
 - ▶ Click "For CD installations, automatically start Setup when CD is inserted"
 - ▶ Click Publish Now
 - ▶ (demo: 16.1_02ClickOnce.wmv)



9



Windows Installer deployment

Windows Installer Deployment

- ▶ It allows you to package the application in a setup.exe file and distribute that file to users, who run the Setup.exe file and steps through a wizard to install the application.
- ▶ This is done by adding a Setup project to your solution.
- ▶ Visual Studio deployment uses Microsoft Windows Installer technology to create installers that give you complete control over the installation process.

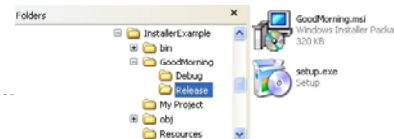
Windows Installer

- ▶ each computer keeps a database of information about every application that it installs, including files, registry keys, and components.
- ▶ When an application is uninstalled, the database is checked to make sure that no other applications rely on a file, registry key, or component before removing it. This prevents the removal of one application from breaking another.

10



Deployment tools



The deployment tools

- ▶ can be used to create Windows Installer (.msi) files that
 - ▶ can be distributed on traditional media such as floppy disks or CD-ROM, or on a networked drive for installation across a network,
 - ▶ and installed files into a Program Files directory on other computers.
- ▶ include the capability to automatically detect the existence of components during installation and install a predetermined set of prerequisites — a process known as bootstrapping.
 - ▶ Most applications have prerequisites:
 - Components such as the .NET Framework runtime must be available on a target computer in order for the application to run
 - ▶ For Windows Installer deployment, bootstrapping of the .NET Framework is enabled by default.

11



Application Creation & Deployment

Steps:

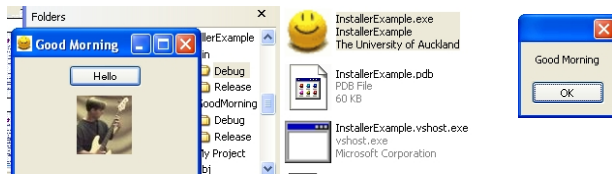
- ▶ Windows Application
 - ▶ Create a Windows-based application
 - ▶ Configure the Windows Forms Application
 - ▶ Compile a Release build of the Windows Forms Application
- ▶ Deployment:
 - ▶ Create a Deployment project within the solution
 - ▶ Configure the Deployment project
 - Specify the files that you want to deploy
 - Specify the items that you want added to the Start menu on the target computer
 - ▶ Build the Deployment project thus creating the Setup.exe file and related files
 - ▶ Distribute the Setup.exe file and related files
- ▶ Note: a Setup wizard is provided to help step you through the process of creating deployment projects.

12



Example

- ▶ **Windows Application:**
 - ▶ Create a new Windows Application project
 - ▶ Implement the programming logic
 - ▶ Set up an Application Icon
 - ▶ Set up a Form Icon
 - ▶ Compile a Release build of the application
 - ▶ (demo: 16.1_01AddIconAndRelease.wmv.)



13

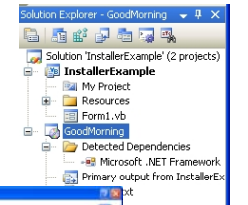
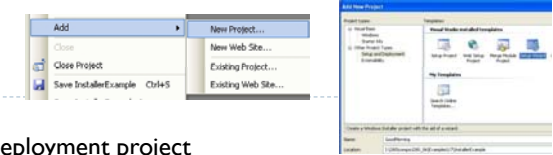
COMPSCI280

Handout16

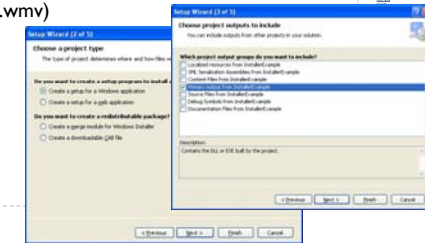


Deployment

- ▶ To create a new Setup and Deployment project
 - ▶ On the File menu, point to Add, then click New Project.
 - ▶ In the Project Types pane, open the Other Project Types node and select **Setup and Deployment** Projects.
 - ▶ Choose Setup Wizard
 - ▶ Specify the type of project to deploy
 - Select "Create a setup for a Windows application"
 - ▶ Specify the program outputs
 - Select "Primary output"
 - ▶ Select any additional support files
 - Select "readme.txt"
 - ▶ (demo: 16.2_01SetupWizard.wmv)



14



Handout16



Deployment Project Properties

- ▶ Deployment projects have two categories of properties: general project properties and configuration-dependent properties.
 - ▶ General properties are set in the Properties window and apply to all project configurations.
 - ▶ To set configuration-dependent properties, you use the Deployment Properties dialog box; configuration-dependent properties apply to a specific project configuration.
- ▶ To set the configuration-dependent properties
 - ▶ Select the setup project node.
 - ▶ Choose View | Property Pages.
 - ▶ Output file name
 - Specifies the location where the Windows Installer (.Msi, .Msm) file will be placed when built.
 - ▶ Package files
 - Specifies how files will be packaged within a Windows Installer file.
 - ▶ Prerequisites
 - Specify which prerequisite components to install, and specify the locations from which your application and the prerequisite components will be installed.
 - ▶ Select a configuration Manager
 - ▶ Select Release and click "Build" of the deployment project.

15

COMPSCI280

Handout16



Deployment Project Properties

- ▶ To set the deployment project properties
 - ▶ Select the deployment project and open the Properties window
 - ▶ **Author** and **Manufacturer** properties contain the name of the company that developed the application
 - It is also used as a part of the default installation path (C:\Program Files\Manufacturer\Product Name) displayed during installation.
 - ▶ **Description** property contains a textual description
 - ▶ **DetectNewerInstalledVersion** property will prompt the user if a newer version of the application is already installed
 - ▶ **Title** specifies the title of an installer. By default, the Title property is the same as the name of the deployment project.
 - ▶ **RemovePreviousVersions** property allows the installer to remove old software versions
 - ▶ **Version** property assigns a version number to the installer. This is the version of the installer not the application



16

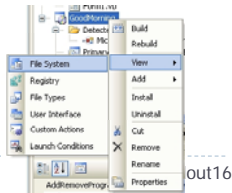
COMPSCI280

Handout16



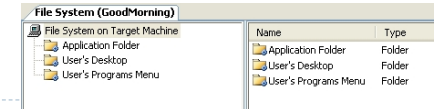
Deployment Project Editors

- ▶ Use the Deployment Project Editors to manually configure the Deployment project
 - ▶ **File System Editor** is used to define which files will be deployed and where on the target computer those files will be deployed
 - ▶ The **Registry Editor** allows you to define an installer that will add or update entries to the Registry on the target computer
 - ▶ The **File Types Editor** creates file associations
 - ▶ The **User Interface Editor** allows you to add custom dialog boxes to the installer
 - ▶ The **Custom Actions Editor** allows you to further customize the installation process
 - ▶ The **Launch Conditions Editor** will check that specific components are installed on the target computer



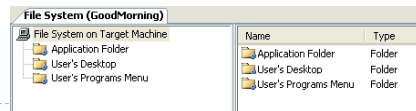
File System Editor

- ▶ Use File System Editor to
 - ▶ Specify the project outputs and other files that you want to deploy to the target computer
 - ▶ Specify any items that will be added to the Start menu on the target computer
 - ▶ Specify any shortcuts that will be added to the desktop on the target computer
- ▶ The File System Editor is divided into two parts: a navigation pane on the left and a detail pane on the right.
 - ▶ The navigation pane contains a hierarchical list of folders that represent the file system on a target computer. The folder names correspond to standard Windows folders.
 - ▶ When a folder is selected in the navigation pane, any files and shortcuts that will be installed in that folder are displayed in the detail pane.



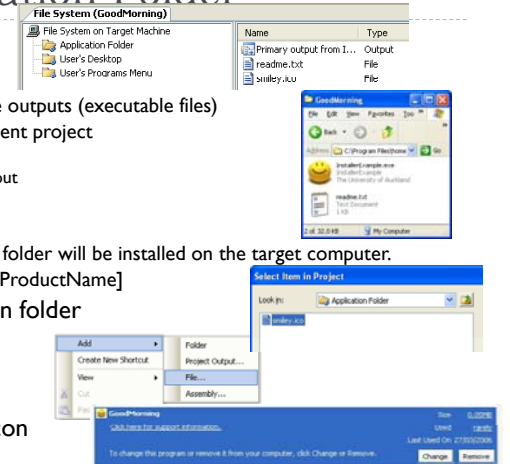
File System Editor

- ▶ To open the File System Editor
 - ▶ Select a deployment project in Solution Explorer.
 - ▶ Right-click and click View | File System menu
- ▶ Three Folders appear in the File System Editor
 - ▶ Application Folder
 - ▶ An application folder under the Program Files folder
 - ▶ User's Desktop
 - ▶ A folder that contains files and folders that appear on the desktop on a per-user basis. Typically C:\Documents and Settings\username\Desktop.
 - ▶ User's Programs Menu
 - ▶ A folder that contains a user's program groups, typically C:\Documents and Settings\username\Start Menu\Programs.
- ▶ Note: New folders can be added to the above folders
 - ▶ Adding a Folder causes the installer to create a new folder on the target computer
 - ▶ Multiple folders and sub folders can be added
 - ▶ (demo: I6.2_02Extra.wmv)



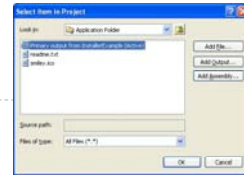
Example: Application Folder

- ▶ Application Folder
 - ▶ It contains a primary output file
 - ▶ Every solution produces one or more outputs (executable files)
 - ▶ To add a project output to a deployment project
 - Select the Application Folder item
 - Right-click and click Add | Project Output
 - Select primary output
 - ▶ Default Location Property
 - ▶ Specifies the default location where a folder will be installed on the target computer.
 - ▶ [ProgramFilesFolder][Manufacturer][ProductName]
- ▶ To add an icon file to the Application folder
 - ▶ Select the Application Folder item
 - ▶ Right-click and click Add | File
 - ▶ Select the file and click Open
- ▶ To add an Add/Remove Programs icon
 - ▶ Select your Setup project.
 - ▶ In the Properties window, select the AddRemoveProgramsIcon property, and choose (Browse...) from the drop-down list.
 - ▶ Click Browse and select the icon file from the Application folder

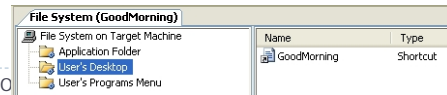




User's Desktop



- ▶ To add a Shortcut in the User's Desktop folder
 - ▶ Select the User's Desktop Folder item
 - ▶ Right-click and click Create New Shortcut
 - ▶ Select the Primary output item from the Application Folder and click OK
 - ▶ We are only allowed to create shortcut on the user's desktop to files that are part of our installation package.
 - ▶ Rename the shortcut
- ▶ To add an icon for the Shortcut
 - ▶ Select the shortcut in the User's Desktop folder
 - ▶ In the properties window, select the Icon property, and choose (Browse...) from the drop-down list.
 - ▶ Click Browse and select the icon file from the Application folder
 - ▶ Select the icon and click OK



21

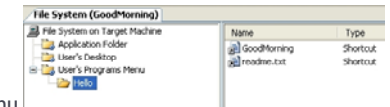
CO

6



User's Programs Menu

- ▶ The User's Programs Menu represents the location on the user's computer inside the Programs folder on the Start menu.
 - ▶ We can add additional folders to the Programs Menu
 - ▶ We can create shortcuts to the program and additional files in the newly created folder
- ▶ To add a Folder in the User's Programs Menu folder
 - ▶ Select the User's Programs Menu Folder item
 - ▶ Right-click and click Add | Folder
 - ▶ Change the name of the new folder
- ▶ To add a Shortcut to the program
 - ▶ Select the new folder in the User's Programs Menu Folder item
 - ▶ Right-click and click Create New Shortcut
 - ▶ Select the Primary output item from the Application Folder and click OK
 - ▶ Rename the shortcut and set the Icon of the shortcut
- ▶ To add a Shortcut to the readme.txt
 - ▶ Select the new folder in the User's Programs Menu Folder item
 - ▶ Right-click and click Create New Shortcut
 - ▶ Select the readme.txt item from the Application Folder and click OK
 - ▶ Rename and set the Icon of the shortcut



22

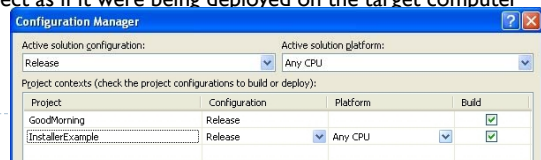
COMPSCI280

Handout16



Building the Setup Package

- ▶ You build the Deployment project just as you must build any other project. The build produces the following files:
 - ▶ xxx.msi contains the installer
 - ▶ Setup.exe to run the installer
- ▶ To build the deployment project
 - ▶ In the Configuration Manager box, select Release and click "Build" of the deployment project.
 - ▶ Click OK.
- ▶ To test the installer
 - ▶ Select the the Deployment project in the Solution Explorer
 - ▶ Right-click and click Install
 - ▶ This will deploy the project as if it were being deployed on the target computer



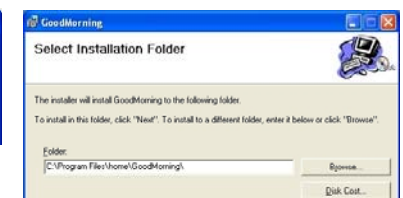
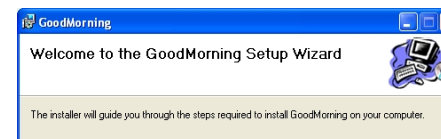
23

16



Default Installer Dialog Boxes

- ▶ When the user runs the installer the following dialog boxes appear by default
 - ▶ Welcome dialog box displays a welcome message
 - ▶ Installation Folder allows the user to override the default installation folder
 - ▶ Confirm Installation dialog box allows the user to confirm or abort the installation
 - ▶ Progress dialog box displays the relative progress of the installation
 - ▶ Finished dialog box displays an installation summary



24

COMPSCI280

Handout16