

Introduction to .NET Deployment

Brian Noyes

IDesign, Inc. (www.idesign.net)

brian.noyes@idesign.net

About Brian

- Principal Software Architect, IDesign Inc. (www.idesign.net)
- Microsoft MVP in ASP.NET
- Writing
 - MSDN Magazine, asp.netPRO, Visual Studio Magazine, .NET Developer's Journal
 - Building Windows Forms Data Applications with .NET 2.0, Addison-Wesley, expected release spring 2005
- Speaking
 - Microsoft TechEd, Visual Studio Connections, DevEssentials, VSLive!, INETA Speakers Bureau
- Participates in Microsoft design reviews
- E-mail: brian.noyes@idesign.net
- Blog: <http://www.softinsight.com/bnoyes>

Agenda

- Deployment Challenges
- Deployment Models
- No Touch Deployment
- Setup and Deployment Projects
- Custom Installers

Deployment Challenges

- Platform requirements
- Intuitive and easy installer UI
- Application artifact placement
 - Files / Folders
 - Registry
 - Global Assembly Cache
 - Databases
- Custom install procedures
- Removing installation artifacts at uninstall
- Handling Errors

Deployment Options in .NET

- xcopy deployment
- No-Touch Deployment (NTD)
- Updater Application Block
- Windows Installer
- .NET Framework 2.0: ClickOnce

xcopy Deployment

- Process
 - Copy application files to appropriate folder
 - Run

xcopy Deployment

- Benefits

- Easiest deployment model
- Limited potential for interference with other applications at install
- Limited potential for interference with other applications at runtime
- Side-by-side versioning achieved through physical separation

xcopy Deployment

- Drawbacks

- More files on disk
- No shared physical implementation between apps
- Less centralized management
- Need to ensure all files are compatible versions

No Touch Deployment (NTD)

- Otherwise known as
 - Auto-deployment
 - Zero deployment
 - href-exe deployment

No Touch Deployment (NTD)

- Process

- Place application files on deployment (web) server
- Provide URL to EXE to users
- Users click on URL to launch
- EXE and supporting DLLs downloaded and cached on user machine
- Execute in .NET runtime secure sandbox
- Based on Code Access Security for the deployment server location
- Deployment server is checked for new versions every time application is launched

Demo

No Touch Deployment

No Touch Deployment (NTD)

- Benefits

- Easy deployment of smart client applications
- No need for admin to touch client workstations for install
- Auto-updating of application files based on deployment server versions
- Protected from malicious code execution
- Easy, Web-like user experience

No Touch Deployment (NTD)

- Drawbacks

- Only applicable to client Windows applications
- Security protections can limit application capabilities
- Must be online to execute, no offline use
- No control over download/update process
- Must resort to workaround approaches to overcome the other drawbacks

Updater Application Block

- Auto-updating of installed smart client applications from a deployment server
- Process
 - Deploy updates to server with manifest
 - Run controller on client to monitor for changes on server
 - Controller downloads, validates, and optionally post-processes updates
 - App launcher runs new version on next app launch

Windows Installer

- Application installation and configuration service
- Manages installation and removal of applications
- Centrally defined rules and repository
- Provides disaster recovery and rollback mechanisms

Windows Installer

- Highly customizable process and database of installation steps and items
- Uses Installer deployment packages (.msi files)
 - Executable format, often called by setup.exe
- Current version 3.0
- Version 2.0 used by Visual Studio 2003 / .NET 1.1
 - Supported OS: Win 95, 98, ME, NT4, 2K
 - Part of WinXP, Server 2003

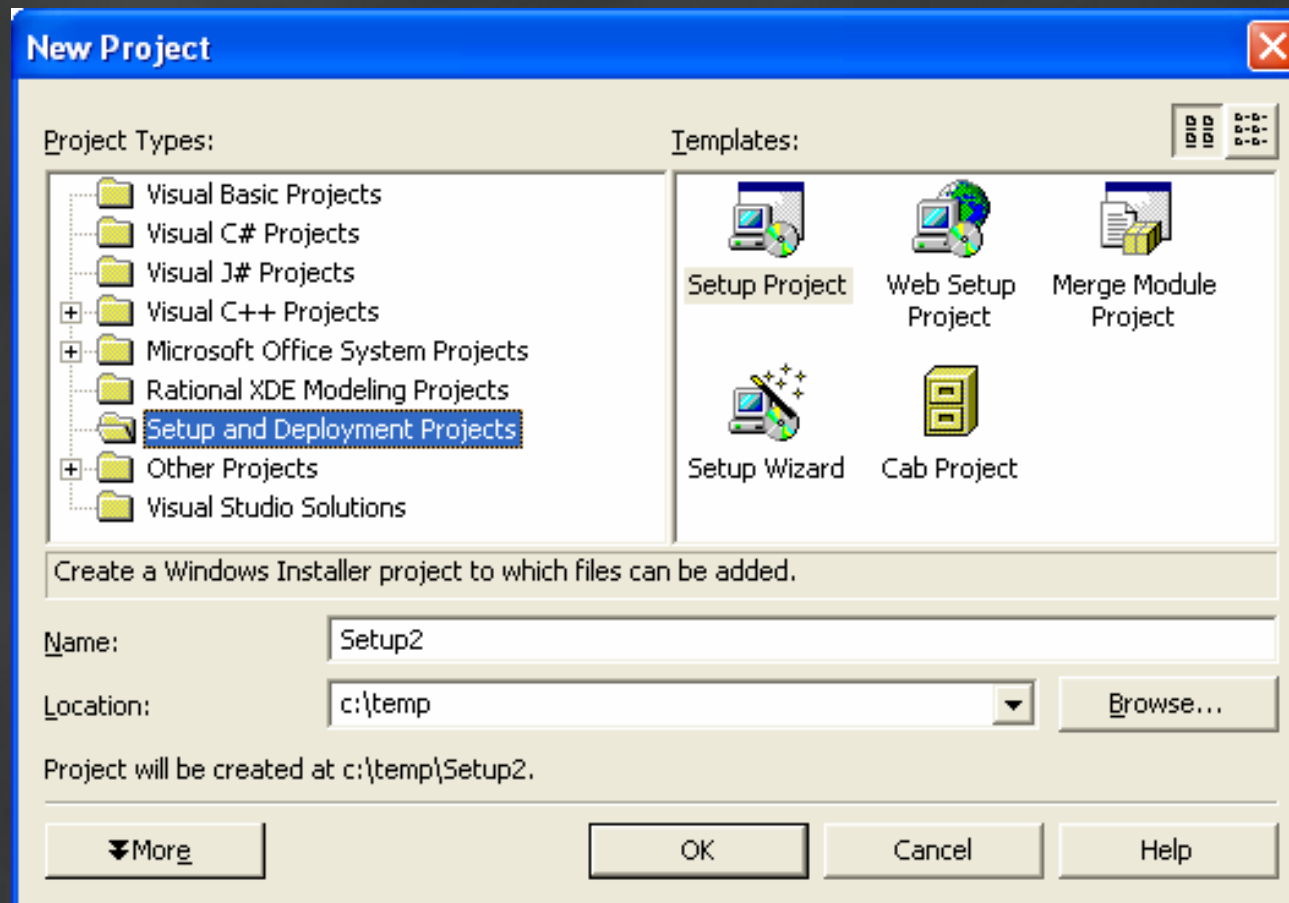
Setup and Deployment Projects

- Built-in project types in Visual Studio.NET
- Creates a Windows Installer package (.msi file)
- Integrates with deployed project through VS.NET solutions
- Fine grained control of installation steps
- Supports Authenticode certificate signing
- Need to have .NET Framework on target machine

Project Types

- **Windows Application**
 - High impact installations on user workstation
- **Web Application**
 - ASP/ASP.NET Web application deployment
 - Creates virtual directory in IIS
- **Merge Module**
 - Reusable installer package to include in other installer package projects
 - Shared Assemblies
- **CAB Project**
 - Legacy packaging of ActiveX controls for web deployment

Setup Project Types



Setup Wizard

Setup Wizard (2 of 4)

Choose a project type

The type of project determines where and how files will be installed on a target computer.



Do you want to create a setup program to install an application?

- Create a setup for a Windows application
- Create a setup for a web application



Do you want to create a redistributable package?

- Create a merge module for Windows Installer
- Create a downloadable CAB file

Cancel

< Back

Next >

Setup Wizard (3 of 4)

Choose files to include

You can add files such as ReadMe files or HTML pages to the setup.



Which additional files do you want to include?

C:\temp\WordAddin\bin\WordAddin.dll
C:\temp\WordAddin\bin\WordAddin.tlb

Add...

Remove

Cancel

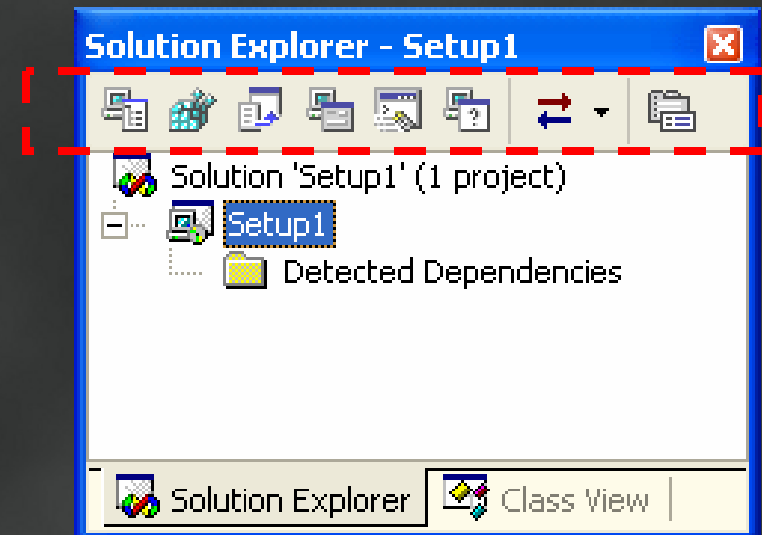
< Back

Next >

Finish

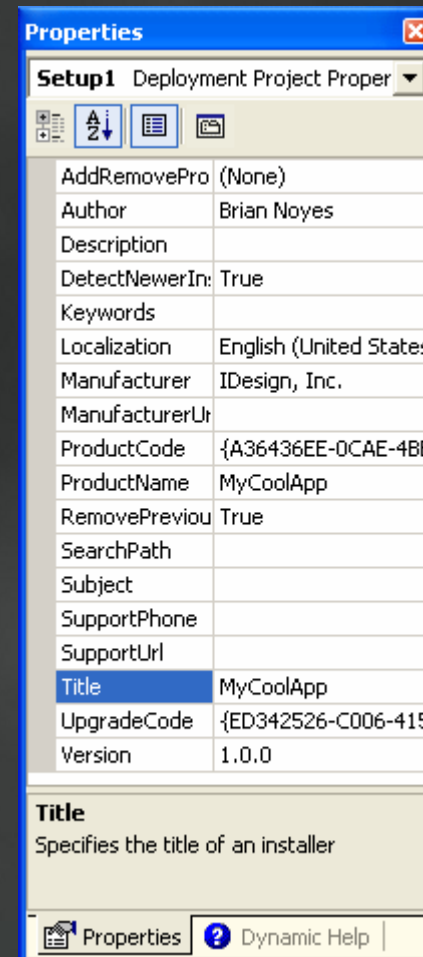
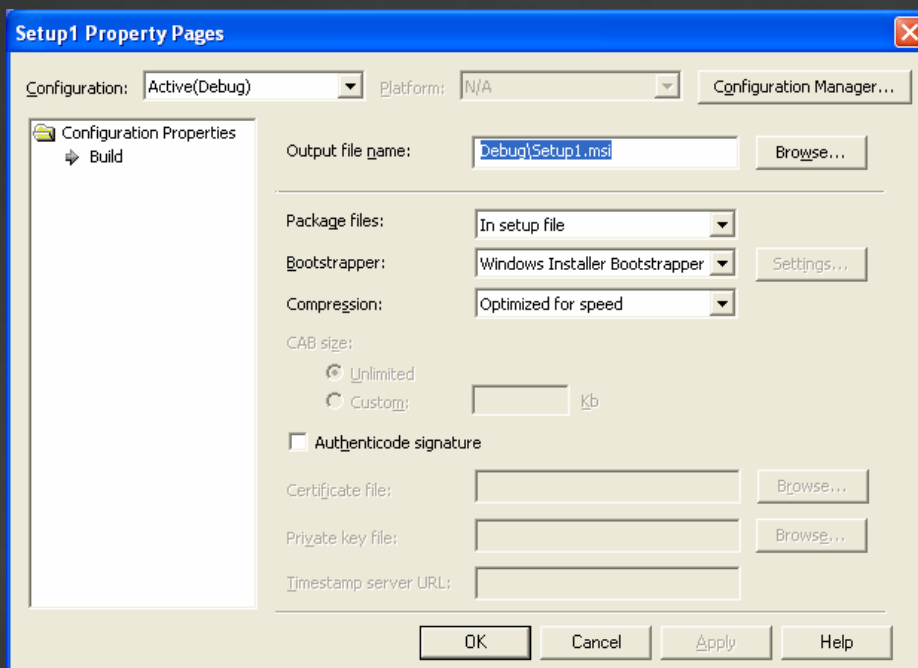
Customizing Setup Projects

- Fine Grained Setup Options
 - File System Editor
 - Registry Editor
 - File Types Editor
 - User Interface Editor
 - Custom Actions Editor
 - Launch Conditions Editor



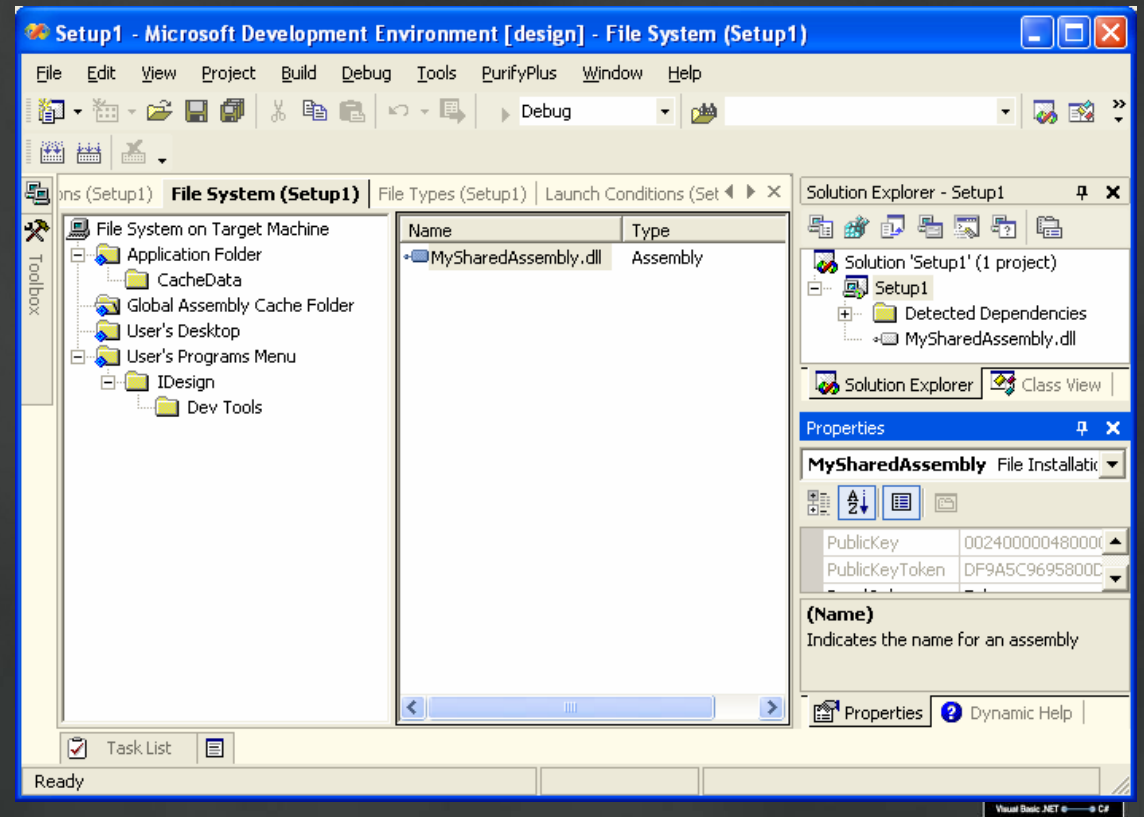
Customizing Setup Projects

- Project Properties
 - Properties Window
 - Property Pages



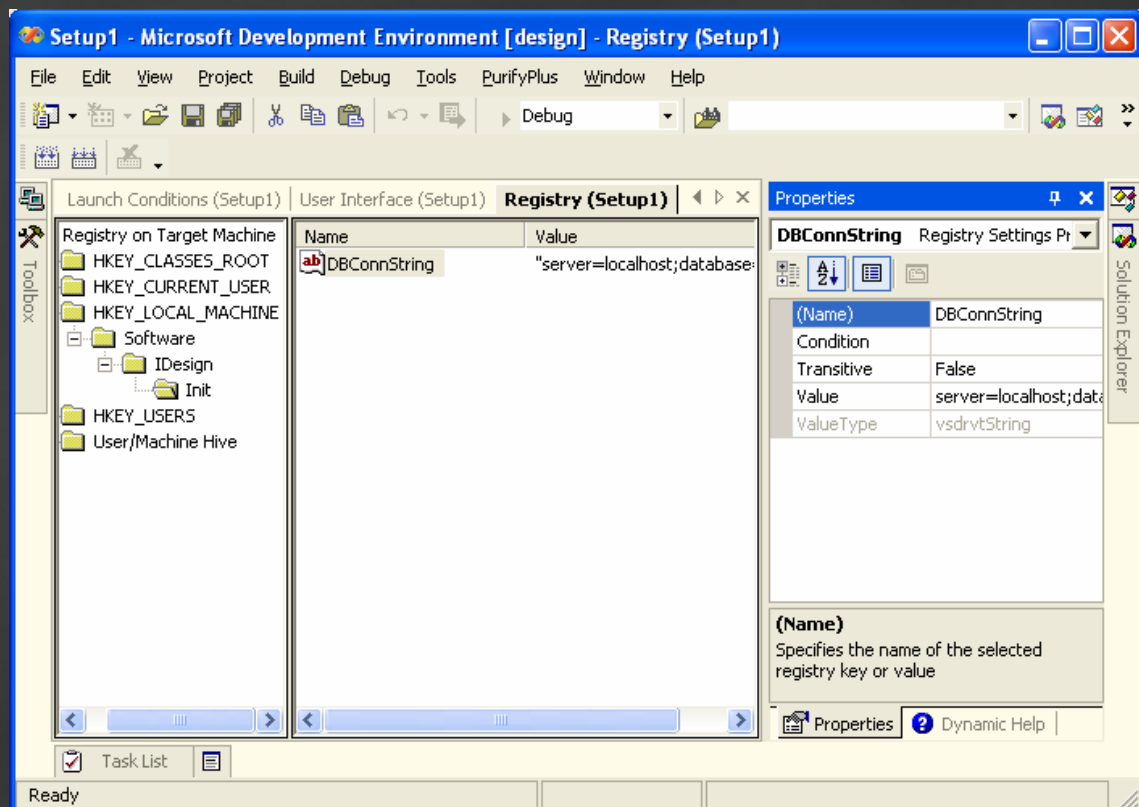
File System Editor

- Create Folders
- Add Files
- Register Shared Assemblies
- Register COM Libraries
- Create Start Menu Shortcuts
- Special Folder Placeholders



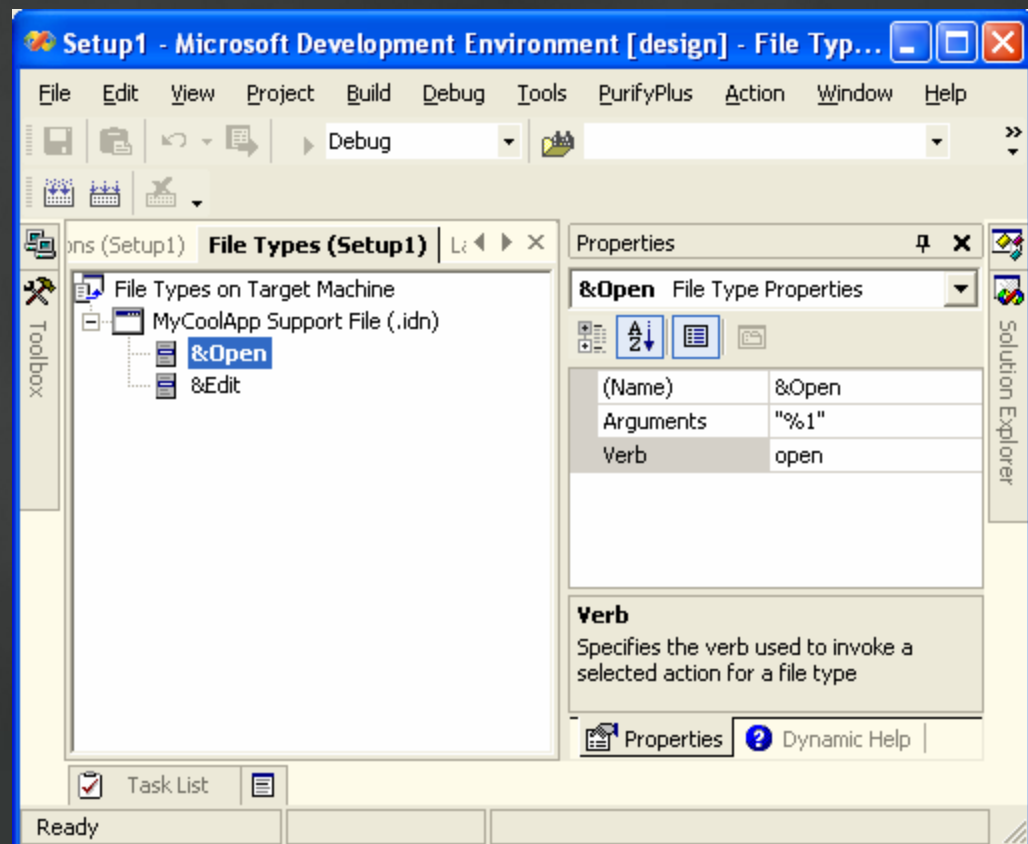
Registry Editor

- Create Keys
- Create Values
- Set Values



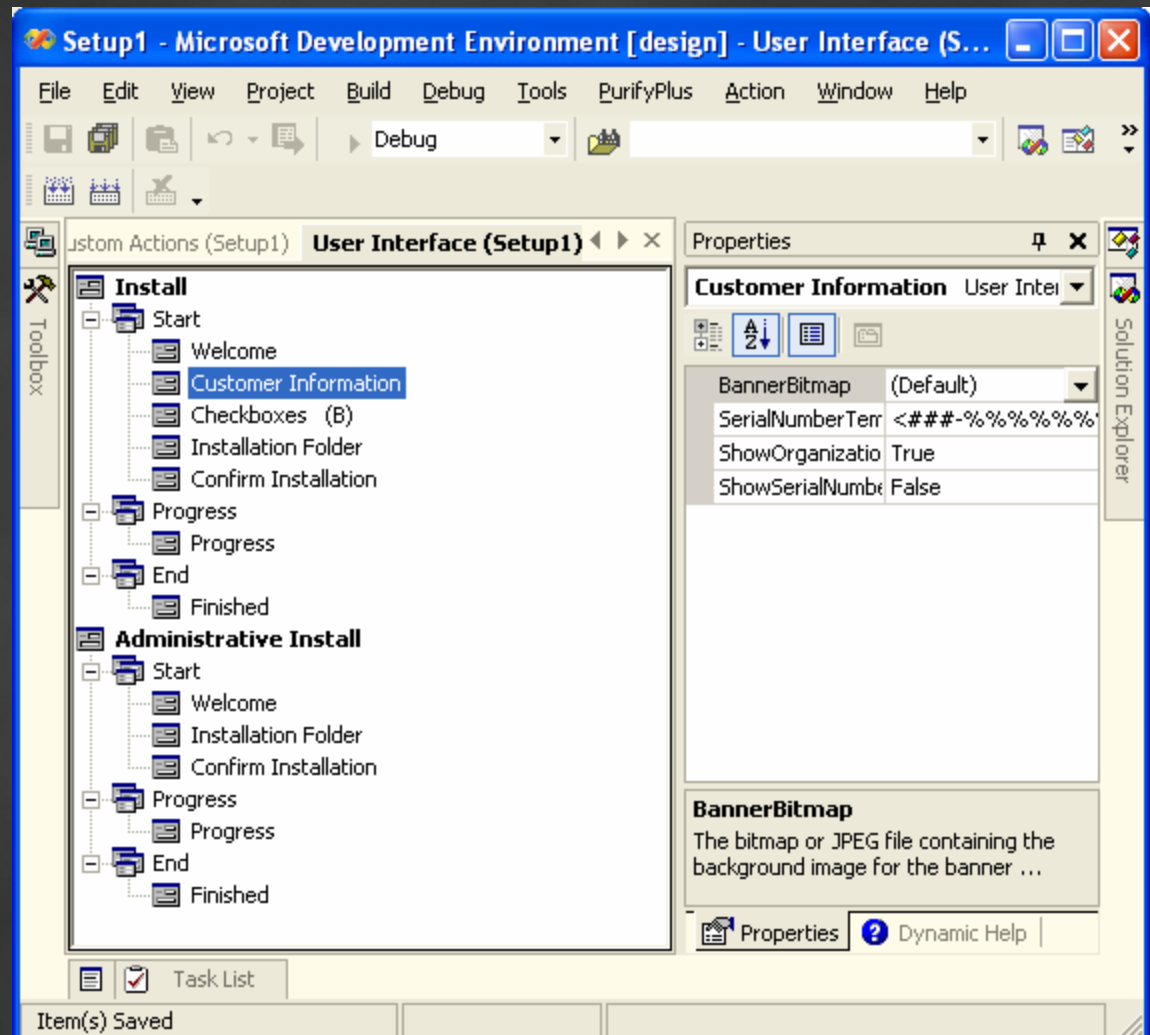
File Types Editor

- Associate File Types (extensions) with your app
- Set OS actions with verbs and arguments



User Interface Editor

- Create different install types
- Customize UI for each install type
- Add canned dialogs
- Add custom dialogs from templates
- Set variables to gather user input
- Somewhat limited customization
- Banner images: 500 X 70 px

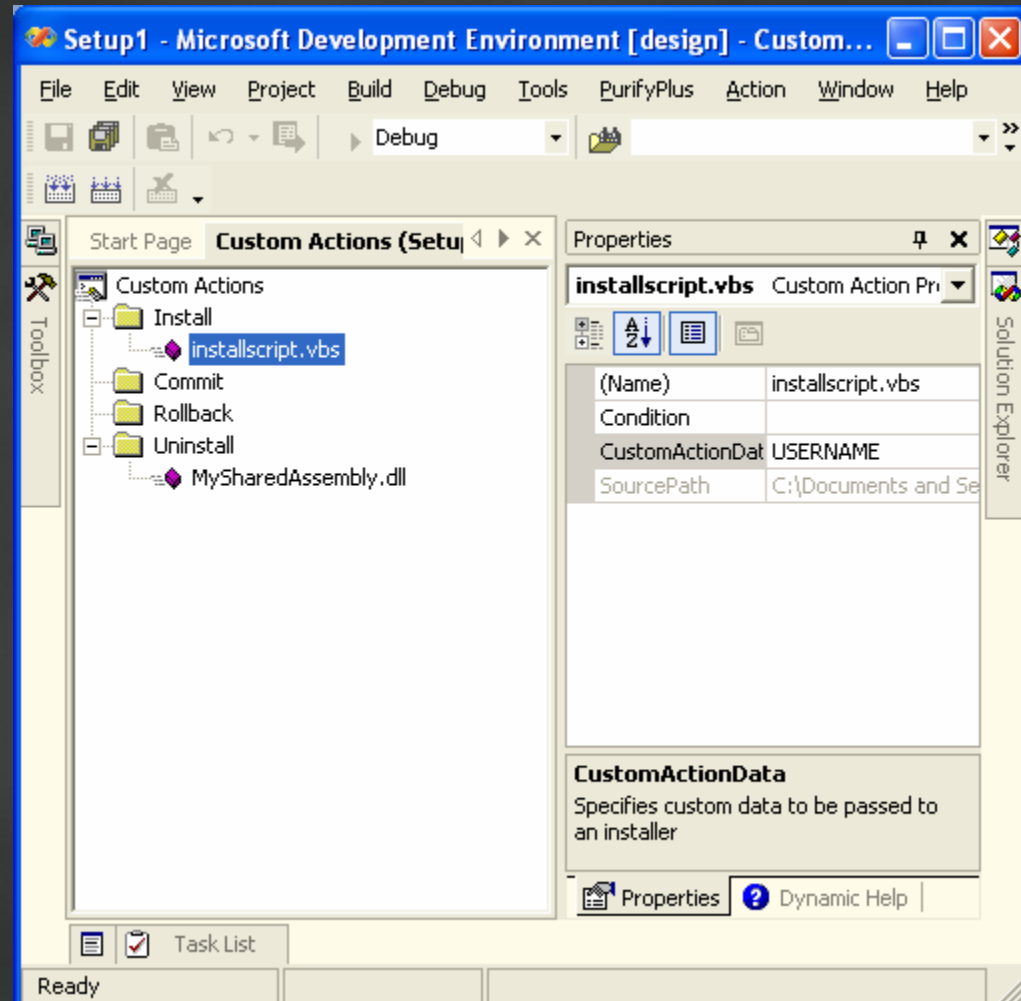


Demo

WinForms Application Deployment

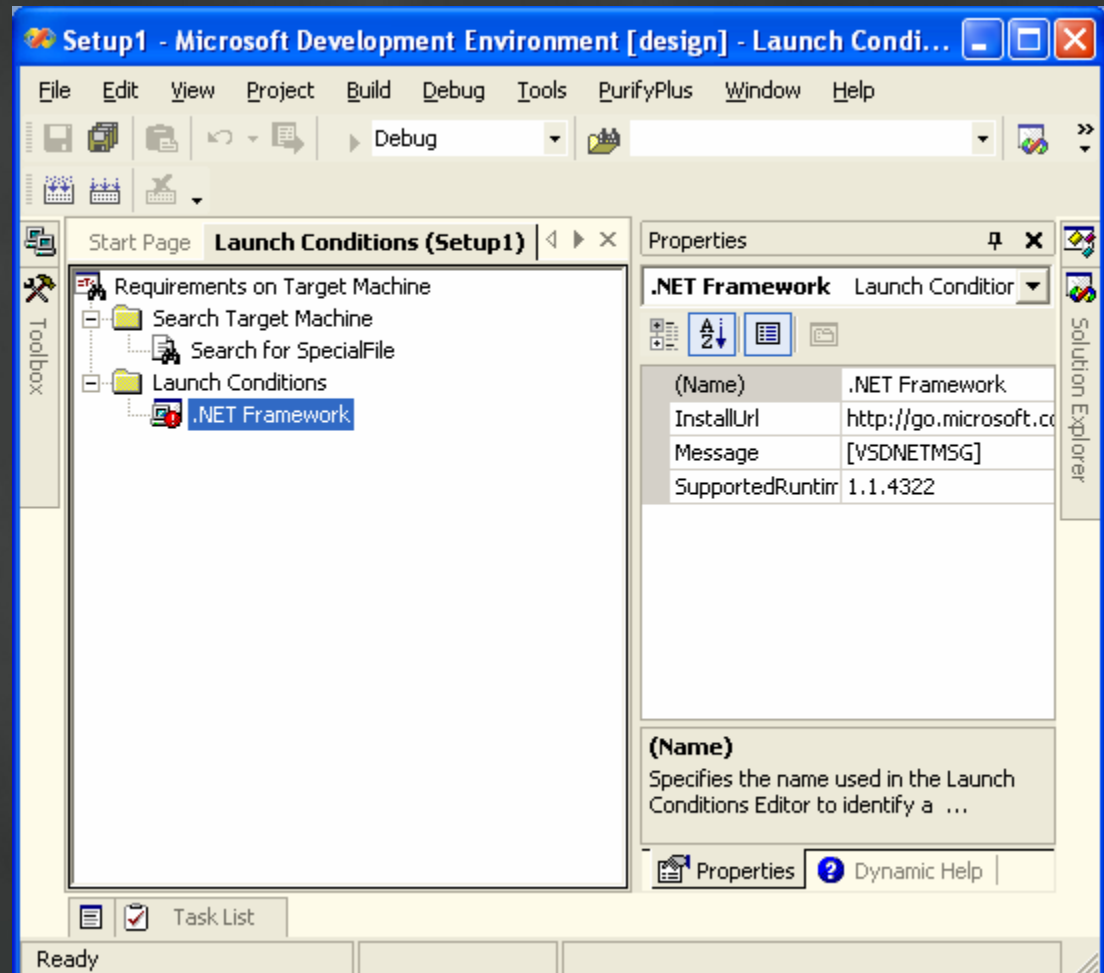
Custom Actions Editor

- Add custom installer scripts, batch files, executables, or .NET Framework Installer assemblies
- Specify arguments to be passed
- Provide Windows Installer conditional statements
- Set different actions for each stage of the process
- Action files must be part of the installation



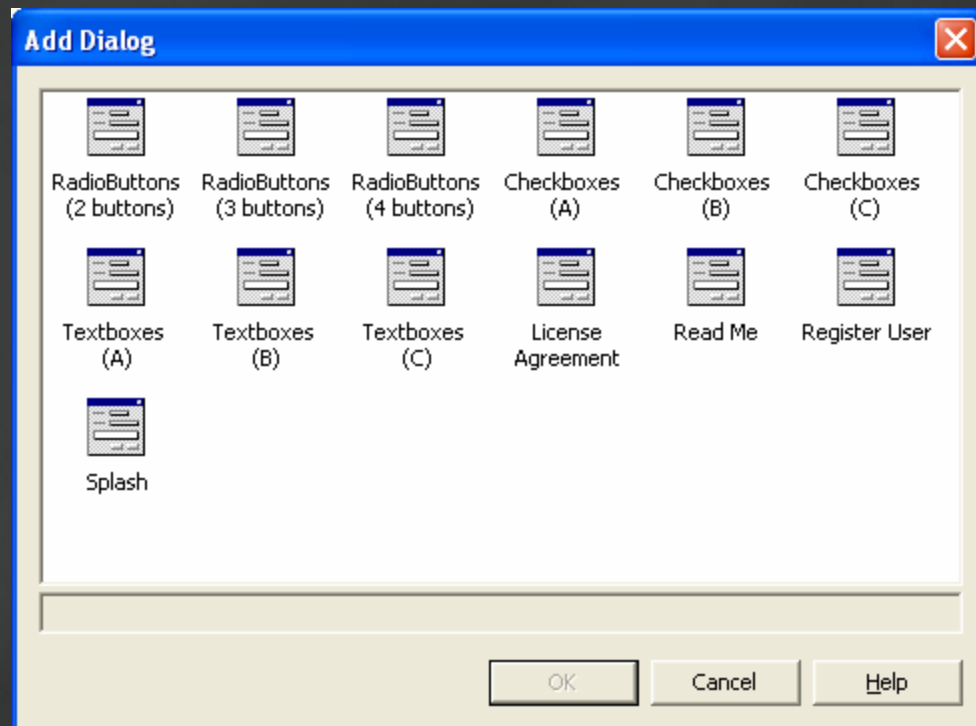
Launch Conditions Editor

- Add launch conditions using Windows Installer condition syntax
- Check for required OS, environment variables, Framework versions
- Check for required files, registry keys, installed components



Creating Custom Dialogs

- Select from predefined set of template dialogs
- Auto-formatted to look like part of the installation
- Various combinations of user input dialogs
- Capture inputs in installer variables
- Can only be inserted at specific points in the install process
- Only one dialog of each type allowed in an install



Capturing User Input

- Pre-defined dialogs
 - Assign variable names to input controls
 - Can pass variables to custom steps or use in conditionals
- Custom Actions
 - Present your own dialogs after completion of the canned dialogs
 - Gather input as desired

Defining Custom Actions

- Custom actions only executed after setup project steps
- Action types
 - Executables (*.exe, *.dll)
 - Scripts (*.vbs, *.js)

Defining Custom Actions

- Add files to File System Folders
 - Must be physically extracted to machine to be executed
- Add custom actions to installation phase node
 - Install, Commit, Rollback, Uninstall
- Pass parameters by property name
 - Collected from user
 - Standard Windows Installer properties

Demo

Custom Actions

Creating Managed Installers

- Derive from Installer base class
 - System.Configuration.Install
- Apply [RunInstaller(true)] attribute
- Place in class library assembly
- Either:
 - Add to custom action
 - Run InstallUtil.exe

Handling Errors

- Rollback method
 - Call during the Install phase if an error occurs
- InstallException
 - Throw during the commit, rollback, or uninstall phases if an error occurs

Summary

- Multiple deployment models
- Smart Client options evolving
- Easy to support complex installation requirements with Setup and Deployment Projects
 - Support managed and unmanaged project installation
- Can create custom managed installers with .NET
- Choose the minimum impact install possible
- Plan for deployment early in requirements analysis and design process

Resources

.NET Deployment

Deploy Apps with Ease, Brian Noyes, asp.netPRO Magazine, Jan 2004

http://www.aspnetpro.com/features/2004/01/asp200401bn_f/asp200401bn_f.asp

No Touch Deployment

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dv_vstechart/html/vbtchno-touchdeploymentinnetframework.asp

Updater Application Block

<http://msdn.microsoft.com/library/en-us/dnbda/html/updater.asp>

ClickOnce Deployment

.NET Rocks! Interview

<http://www.franklins.net/fnetdotnetrocks/dotnetrocks.aspx?showid=48>

Deploy and Update Your Smart Client Projects Using A Central Server, Brian Noyes, MSDN Magazine, May 2004

<http://www.msdn.microsoft.com/msdnmag/issues/04/05/ClickOnce/default.aspx>

Microsoft Installer SDK

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/setup/windows_installer_start_page.asp