Windows Installer for Developers

Introduction

Installing new software is perhaps the most adrenalin-inducing experience you'll have on a computer, aside from whatever games you might play. It's not hard to see why. You give over control of the system to a program which often demands Administrator privilege and which then starts updating some of the most fragile parts of your system. You may know the actual product being installed quite well, but there's rarely any documentation about what the installation of it will do to your system. It might install kernel drivers, services, it might alter your personal settings without your permission, and it might result in other applications on your system no longer working. For a company building and shipping software, the installation might be the first time the customer has seen your product or your company, and it's your opportunity to make a lasting impression one way or the other. An unreliable installation will affect the customer's image of you for a long time.

The goal of this book is to show you how to build safe and secure installations. We'll focus on Windows Installer technology and the Windows NT operating systems, Windows 2000 and above, and example projects will mostly be built as Visual Studio Setup projects. Aside from the actual nuts and bolts of building Windows Installer-based installations, there'll be advice on how to build a reliable installation and what you should and shouldn't do. The integration of installer technology as part of the Windows operating system means that the dividing line between an application and its installation has become much less sharp, and the book will cover how applications need to be designed to properly integrate with Windows Installer.

The book starts with basic principles and drills down deeper in later chapters. We start with the installation equivalent of the "Hello World" program, then gradually get deeper into the contents of installer MSI files, and include installation in the .NET Framework world. Along the way we'll stop to look at best practices and how to keep your installation reliable.

Where use of the installer APIs is being shown, I'll use VBScript for the sake of simplicity and clarity, but I'll also point you at the Win32 equivalents and show you a couple of ways to call them from the .NET Framework language C#. There'll also be some VB.NET samples thrown in for good measure.

Content

Chapter 1
Installations past present and future.
Background. Issues with installations and how the Windows Installer service addresses them. The transactional nature of an install process. Avoiding custom code for routine installation tasks.

Chapter 2

Creating your First MSI file and a Peek Inside it.

Using Visual Studio Installer to build a setup to install notepad.exe and a text file.
An examination of resulting MSI file with Orca (the Windows Installer SDK tool). The File, Feature, Component, and Property tables, and shortcuts. A first look at properties.

An introduction to the Product code, Component code, Upgrade Code, Package Code GUIDs. An introduction to the Windows APIs dealing with MSI files and a VBScript sample to query the contents of our generated MSI file.

Chapter 3

A Look at Sharing using COM

Adding a COM server to our Notepad installation. COM server installation from the installer tables and not with self-registration: Class, ProgID, TypeLib, Registry, and the tables that set up the COM registry entries.
Sharing – COM servers are shared. Where to install a COM server. About the overwrite rules and versions and what a COM interface contract means in this context. Windows Installer sharing using installer components and comparing with the SharedDlls mechanism.
Formal sharing and the idea of Merge Modules as a collection of installer components. Building a COM server as a Merge Module. Side-by-side COM installations.

Chapter 4

Searches and Conditions

Visual Studio's Launch Conditions and how they use AppSearch. Condition on installing components and another look at standard installer properties. Using installer properties as conditions. The installer tables used to check that the system is ready for your setup, LaunchCondition, AppSearch, and the locator tables.
Features as collections of components – Visual Studio Installer does not do multiple features. Transitive component.

Chapter 5

The Order in which Things Happen and Custom Actions

A closer look at the sequence tables and the flow of the installation process, the user interface sequence, the execute sequence, the rollback script. Logging an installation and a first look at what Windows logs in an installation log, including the sequences and properties.
A first look at custom actions and how to insert your own code into the installation process. In-script and deferred custom actions. Creating your own properties. Custom actions as VBScript, calls into a custom Dll, and running EXE files. Communicating with custom actions and CustomActionData. The limitations of Visual Studio Installer in the area of custom actions and adding our own with Orca.

Chapter 6

OK It's Shipped – How Do We Send Out Fixes?
How to update products installed with Windows Installer. How the Product, Upgrade and Package code GUIDs work and how to use them to update a product. Major Upgrades and Minor Upgrades.

Chapter 7
Visual Studio Web Setup Projects
Installing Web applications and Web Services. A Visual Studio Web setup project and how it works.

Chapter 8
Installing .NET Assemblies
Installing assemblies using Windows Installer. The GAC, private assemblies. The MsiAssembly and MsiAssemblyname tables.

How to install assemblies as servers to legacy COM client programs. Where regasm.exe fits in. Side-by-side COM server assemblies.

Chapter 9
Installation Design Issues
A break from the details to look at installation design.

Prerequisites for an Installation. What should the installation check for? What is the application's responsibility? Should the installation require service packs? Should it require special hardware or drivers that are used by the product? Good practice for all installations. What exactly is a Product and why should I worry about it?

How much should the installation do? Should it install databases? What about the user interface and configuring the product? Licenses? Readme files? The transacted nature of an installation and why you should use it wherever possible. The impact of silent installations.

Chapter 10
Windows Services

The ServiceInstall and ServiceControl tables. The issues with Starting, Stopping Services. Service Dependencies. A sample Win32 Service installation with Orca. What happens if there are open Service Handles while you're uninstalling an installing.

.NET Services and Installer-Related Classes
Windows Services written with the .NET Framework and the special custom actions for Install, Uninstall etc. The interaction with the Installer classes and other framework classes called by installation programs.

Chapter 11
Updates in the .NET world
The GAC as a separate file system. Policy files. More on the GAC and updating assemblies in the GAC. AssemblyVersion and the GAC. Gacutil.exe vs. the Windows Installer. Policy files.

Chapter 12
Patches and Transforms.
Building a patch file using the Windows Installer SDK and applying it to an installed product. MSI 3.0 and patches.
Building a transform file with the SDK tools an Orca.

Chapter 13
Installation Environments
Advertised installs. Installing from the Internet. 64-bit installs.

Chapter 14
Tips Tricks Advice ad Info
A grab-bag of advice, warnings, and "how-to".

Chapter 15
More installer APIs.
How to detect and inventory what's installed on the system, where it is and how often it's been used. How to call the APIs from the .NET Framework languages. The WMI interface using the Win32_Product class.

Chapter 16
Futures and Tools
Other products and tools for building MSI files. The design limitations of VS setup projects. What to expect in MSI 3.0.

http://www.installsite.org/go/msibooks.htm