

Master Mobile Software Installation with InstallShield Developer*

Overview

Software installation across an expanding array of mobile computing platforms is complex. Developers are now tasked with building and deploying software on handheld devices ranging from Intel® StrongARM™ powered Pocket PCs to smart phones using the Intel XScale™ microarchitecture. Although this job has been made more manageable with the wide adoption of Microsoft Windows CE* as a common operating environment, problems still abound as developers are required to deliver a separate set of installable binaries for each supported processor architecture.

Fortunately, life is about to become simpler due to the recent release of InstallShield Developer* version 8 and Microsoft's .NET Compact Framework* for Windows CE devices. InstallShield Developer offers a comprehensive installation authoring solution for smart devices and desktop applications. It fully supports creating installations in Visual Studio .NET 2003* including integration and distribution of the Microsoft .NET Compact Framework. InstallShield Developer also has the ability to target all Windows CE platforms, including Windows CE .NET, and other common smart device platforms.

The Challenges of Installation

Even though InstallShield Developer makes the installation problem look easy, the reality is that getting software installation right is an arduous and error-prone task--and the main reason why applications like InstallShield Developer are so popular. Much of the difficulty stems from the nature of Windows CE itself due to its reliance on the much maligned and poorly understood process of creating a CAB file. The CAB file contains detailed instructions telling the Windows CE installation engine exactly where to copy application files and precise directions for modifying system resources.

Developer sentiment towards CAB files is clearly expressed by Art Middlekauff, Software Architect at InstallShield, "Those that have attempted to create CAB files manually are the biggest fans of InstallShield. Even the few hardy developers who have managed to successfully create CAB files often give up in the long run. The task of doing it manually is so complex that during the cycles between product releases developers often forget how the CAB file works--let alone how it can be modified to include the new application binaries." InstallShield Developer radically streamlines this process by providing intuitive Smart Device* and Windows CE wizards that can rapidly create reliable installations for standalone smart device applications or combined desktop/smart device applications.

Supporting a Mobile World

Common to most managed code installations is the need to install the .NET Compact Framework onto the device. The .NET Compact Framework supports a variety of processor architectures which are distributed in separate CAB files that are approximately 2.5MB each.



In addition to packaging the application for installation, InstallShield Developer can bundle in as many .NET Compact Framework CAB files as deemed necessary. This flexibility gives developers the ability to manage the total size of the software distribution --potentially important if the software is to be distributed using fixed capacity storage cards.

As there is no such thing as a single type of smart device, there are multiple ways for installing software onto a smart device.

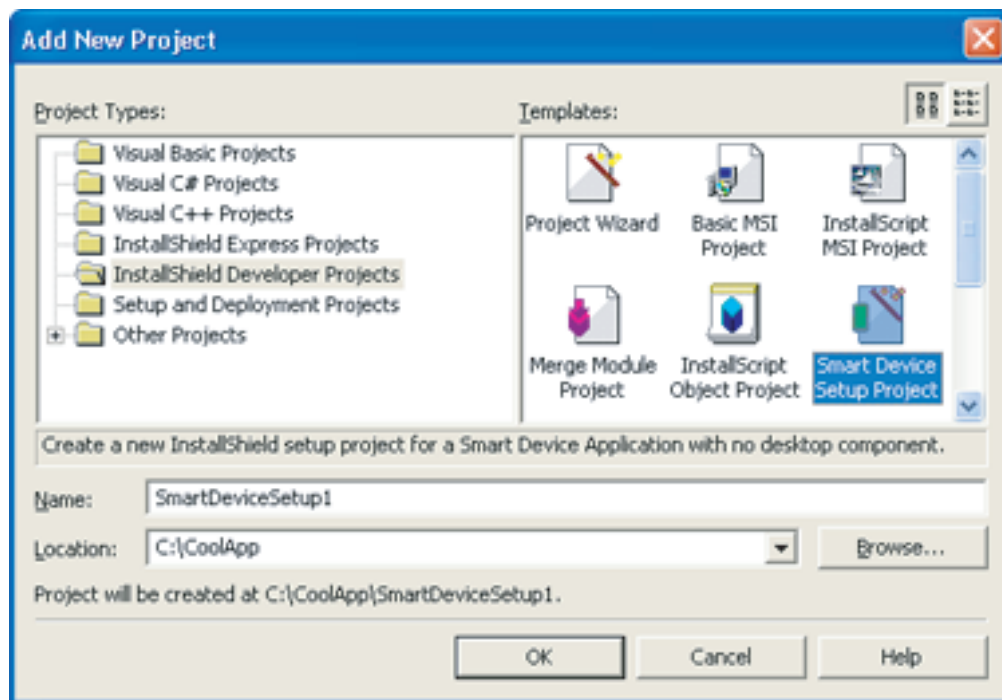


Figure 1. InstallShield Developer 8 is fully integrated into Visual Studio .NET 2003, and includes a Smart Device Setup Project type that can be selected from the Add New Project dialog. Additional Figures: <http://www.installshield.com/isd/info/smart.asp?source>

The majority of handheld devices in use today still connect occasionally to the network using docking stations where they are updated using Microsoft ActiveSync*. Newer models offer wireless connectivity which facilitates over-the-air software installation, while others install software from Compact Flash and Secure Digital storage cards. InstallShield Developer takes into account all these scenarios and supports streamlined installations no matter how software is transferred to the device.

With a tethered approach to software installation the user docks the device in a cradle, starts software installation using the desktop, and relies on ActiveSync to transfer the required CAB files to the device. Supporting this scenario is especially tricky as writing code that can run on the desktop and do a seamless integration with ActiveSync is challenging.

InstallShield Developer makes this easy by encapsulating the functionality in an easy to use wizard. Developers simply choose what files to distribute, decide whether to include the .NET Compact Framework, and wait for the resulting SETUP.EXE file.

Storage cards are proving popular for software distribution in environments with many handheld devices and only a few docking stations--common in warehouse and factory environments.

Devices that support storage cards can make use of the Windows CE facility that automatically runs the AUTORUN.EXE when the card is first inserted. InstallShield Developer simplifies this process by creating the required AUTORUN.EXE file along with CAB file bundles for the application and .NET Compact Framework. The resulting distribution can be copied to a storage card and passed from device to device.

Recent technological advances have furthered the trend towards handheld devices sporting wireless access using technologies such as 802.11b and a variety of 3G wireless WAN standards. With wireless support, users can update their devices from any location with over-the-air installation.

Developers can take the same CAB files that are built for tethered or storage card installation and post them on a Web server. All users have to do is access the correct URL, select the right CAB files, and wait for Windows CE to complete the install. Processor-specific .NET Compact Framework CAB files can also be posted for easy installation.

Unlike traditional Windows where shared DLLs are installed in %windir%\system32 directory, .NET uses the concept of assemblies. An assembly is a collection of modules and resources that are part of the same version of an application and are deployed as a single unit.

An assembly can be either private (available only to the application that uses it) or public (available to all applications resident on the machine). Public assemblies are installed in the .NET global assembly cache (GAC) which is a systemwide repository of versioned assemblies. InstallShield Developer properly manages updating the GAC—all one has to do is select a checkbox in InstallShield Developer to make this happen.

Summary

Although the underlying software environment adds layers of complexity with each operating system release, InstallShield Developer makes the task of creating easily installable software simpler than it has ever been. With full support for a wide variety of smart device architectures, support for the .NET Compact Framework, and tight integration with Visual Studio .NET, InstallShield Developer 8 provides a powerful solution for creating installation packages that can be deployed no matter how users choose to update their handheld device.

For additional information on InstallShield Developer, visit <http://www.installshield.com/isd/>

InstallShield®

Pointers from the Pros

- **Redistribute the .NET Compact Framework:**
As the .NET Compact Framework was just formally released by Microsoft, it is not yet present on most devices. If an organization uses a constrained set of handheld devices—say for tracking package deliveries—it is possible to minimize the installation footprint by only including .NET Compact Framework support for those devices.
- **Use InstallShield Developer directly from Visual Studio .NET:**
Although InstallShield Developer includes a standalone IDE, it is preferable to run the product directly from within Visual Studio .NET. The seamless integration with Visual Studio .NET simplifies life as all InstallShield Developer views are presented within Solution Explorer—along with automatically updating the installation package with each build.
- **Use the global assembly cache:**
Developers can lower the memory footprint of an application by installing shared functionality into the global assembly cache. This requires making upfront architectural decisions about what functionality goes into each assembly.

About the Author

Allan Mc Naughton is the principal analyst at Triangle Technology Advisors LLC (<http://www.triangletechadvisors.com>), a firm specializing in the composition of high-technology white papers. He can be reached at allan@triangletechadvisors.com.

Additional Resources

Intel® Developer Services provides online resources to help software developers accelerate the development and delivery of their applications, tools and solutions.

Find out more at the Intel Developer Services website at: <http://www.intel.com/ids>

Information in this document is provided in connection with Intel® products. Except as provided in Intel's terms and conditions of sale for such products, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUALPROPERTY RIGHT. Intel products are not intended for use in medical, life-saving, life-sustaining, critical control or safety systems, or in nuclear facility applications. Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference www.intel.com/procs/perf/limits.htm, or call (U.S.) 1-800-628-8686 or 1-916-356-3104.

Intel, the Intel logo, Pentium and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

