



# InstallShield 2010

## Release Notes

June 18, 2009

(updated to include SP1, released on October 6, 2009)

### Introduction

InstallShield is recognized as the industry standard in installation authoring. Successful software publishers around the world rely on InstallShield for authoring powerful and flexible Windows Installer and InstallScript installations.

InstallShield 2010 offers many new features and enhancements that help you use the latest technologies and save time on everyday tasks. InstallShield now supports Windows 7, Windows Server 2008 R2, Windows Installer 5, and SQL Server 2008 SP1. Many of the views in InstallShield contain new toolbars and dynamic search capabilities that make it easier to find information, modify project settings, and customize grids. A new String Editor view enables you to view and edit all localizable run-time strings in one centralized place. The new Text File Changes view lets you quickly configure search-and-replace behavior for content in text files on target systems. The redesigned Internet Information Services view now lets you create IIS Web applications and virtual directories separately. Plus, you can now add InstallShield prerequisites—which were previously supported only in Basic MSI and InstallScript MSI projects—to InstallScript projects.

InstallShield 2010 Service Pack 1 (SP1) includes changes that offer support for the final released versions of Windows 7, Windows Server 2008 R2, and Windows Installer 5. (Note: Without SP1, InstallShield 2010 supports Microsoft Windows 7 Release Candidate English, Microsoft Windows Server 2008 R2 Release Candidate English, and the beta version of Windows Installer 5.)

For the latest information about InstallShield 2010, including updates to these release notes, see Knowledge Base article [Q200329](#).

**Note:** If you have not purchased a license for InstallShield, you can install it and use it for a limited number of days without activating it. When you use InstallShield before activating it, it operates in evaluation mode, and some of its functionality is not available. For details, see KB article [Q200900](#). Note that when you activate InstallShield, the evaluation limitations are removed.

### New in InstallShield 2010 SP1 (October 6, 2009)

To obtain SP1, see Knowledge Base Article [Q206429](#).

InstallShield 2010 SP1 includes changes that offer support for the final released versions of Windows 7, Windows Server 2008 R2, and Windows Installer 5. It also includes additional changes.

**Upgrade Note:** If you want to open an InstallShield 2010 project in InstallShield 2010 SP1, you must allow InstallShield to upgrade your project to InstallShield 2010 SP1. InstallShield 2010 SP1 includes support for tables that were not available in InstallShield 2010 projects, and these tables need to be added during the upgrade. Note that it is not possible to open InstallShield 2010 SP1 projects in earlier versions of InstallShield (including InstallShield 2010 without SP1). Therefore, if multiple users need to open and modify your InstallShield projects, ensure that all of you apply the SP1 patch at the same time.

If you open an InstallShield 2010 project in InstallShield 2010 SP1, InstallShield 2010 SP1 displays a message box that asks you if you want to convert the project to the new version. If you reply that you do want to convert it, InstallShield creates a backup copy of the project before converting it.

### **Setup.exe Manifests Now Have Compatibility Section to Avoid Triggering Program Compatibility Assistant on Windows 7 and Windows Server 2008 R2 Systems**

If you configure your InstallShield project to create a setup launcher for your installation, the manifest that InstallShield creates for the setup launcher now includes a compatibility section. Previously, without this compatibility section, a Program Compatibility Assistant (PCA) dialog box may have appeared at the end of an installation on Windows 7 and Windows Server 2008 R2 systems. The PCA dialog box indicated that the program might not have installed correctly. This dialog box was displayed if the installation did not create the application uninstallation key. This may happen if the end user cancels the installation or the installation fails to complete successfully.

### **Support for Creating App-V Package Upgrades and for Compressing App-V Packages**

InstallShield now has support for creating App-V package upgrades. The Package Information page in the Microsoft App-V Assistant has a new Upgrade Settings link. Click this link to specify whether you want to create an upgrade. If you specify that you do want to create an upgrade, you can specify additional information about the upgrade, such as whether to append the version number to the App-V package file name. By default, App-V package upgrades are not created.

The Build Options page in the Microsoft App-V Assistant has a new setting that lets you specify whether you want to use compression for the data files in the App-V package. If you select Yes for this setting, InstallShield uses zlib compression for your App-V package.

The App-V upgrade and compression functionality is available in the following project types: Basic MSI and MSI Database.

The Microsoft App-V Assistant is available if you purchase InstallShield with the Virtualization Pack.

### **Windows Installer 5 Support for Configuring New Customization Options for Windows Services, Plus Other Service-Related Improvements**

InstallShield now includes support for configuring extended customization options for Windows services. The customization options include a new delayed auto-start capability to help system startup performance, improved failure detection and recovery options to improve system reliability, and more. Windows Installer 5 supports these new options; earlier versions of Windows Installer ignore them.

To configure the new service-related settings, use the new Services node in the Advanced Settings area for a component in the Setup Design view (in installation projects) or the Components view. All of the services for a component are now grouped and listed by service name under the Services node. In addition, the previously available service-related settings are now consolidated in the same grid with the new settings. In earlier versions of InstallShield, these previously available service-related settings were split among separate subviews for the Install NT Services and Control NT Services nodes. The consolidated grid of settings makes it easier to create a component that installs, configures, starts, stops, or deletes a new or existing service during installation or uninstallation.

This functionality is available in the following project types: Basic MSI, InstallScript MSI, Merge Module, MSI Database, MSM Database, and Transform.

### Validation for the Windows 7 Logo Program

InstallShield includes two new validation suites: InstallShield Validation Suite for Windows 7 and InstallShield Merge Module Validation Suite for Windows 7. These validation suites can help you verify that your Windows Installer–based installation or merge module meets the installation requirements of the Compatible with Windows 7 logo program. If you want to be able to use the Windows 7 logo artwork, your product's installation must meet the program's requirements.

If you want to configure InstallShield to perform validation with these validation suites each time that a release is successfully built: On the Tools menu, click Options. On the Validation tab, select the appropriate check boxes.

If you want to perform validation separately from the build process: On the Build menu, point to Validation, and then click the appropriate new suite.

### Windows Installer 5 Support for Setting Windows Shell Properties for a Shortcut

The Shortcuts view in InstallShield now lets you specify one or more shortcut properties that need to be set by the Windows Shell at run time. For example, if you do not want the Start menu entry for one of your shortcuts to be highlighted as newly installed after end users install your product, you can set a property for that shortcut. You might want to use this property with shortcuts that are for tools and secondary products.

Windows Installer 5 includes support for setting Shell shortcut properties. Earlier versions of Windows Installer ignore those properties.

### Ability to Specify a Custom .png Shortcut Icon for the Start Screen on Windows Mobile 6.5 Professional–Powered Devices

If you are creating a mobile device installation for Windows Mobile–powered devices and you plan to distribute your product through Windows Marketplace for Mobile, you need to use a 90×90 pixel Start screen icon (.png file) for your product. Both the Windows Mobile Wizard and the Smart Device Setup Wizard in InstallShield enable you to select the .png file that you want to use. When the shortcut is installed and displayed on Windows Mobile 6.5 Professional–powered devices, the Windows Mobile Shell scales down the shortcut's icon as necessary, depending on the DPI of the device.

The Shortcut Properties dialog box has a new "Use a Start screen icon (.png)" check box. If you want to use a .png file, select this check box, and then click the browse button to specify the file. The Shortcut Properties dialog box opens when you are using the Windows Mobile Wizard or the Smart Device Setup Wizard and you add a new shortcut or change the properties of an existing shortcut on the Shortcuts panel.

This functionality is available in the following project types: Basic MSI, InstallScript MSI, and Smart Device.

### Windows Installer 5 Support for the MsiPrint and MsiLaunchApp Control Events

When you are adding a control to a dialog in your project, you can select one of the new events that is now available for Windows Installer 5:

- **MsiPrint**—You can add this event to a push button control that is on a dialog that has a scrollable text control. When an end user clicks the push button control, the contents of the scrollable text control are printed.
- **MsiLaunchApp**—You can add this event to a check box control on a dialog, and select the file that you want to be launched in the event's Argument setting. The check box enables end users to choose whether to run the file at the end of the installation. This event is typically used with a check box control that is on the SetupCompleteSuccess dialog. The check box control should include a condition that prevents the control from being displayed during uninstallation.

Windows Installer 5 includes support for these control events. Earlier versions of Windows Installer ignore these events. Therefore, if your installation is run on a system that has Windows Installer 4.5 or earlier and you want to use one or both of these new events, add a condition to the dialog controls so that they are not displayed on systems that have Windows Installer 4.5 or earlier.

This functionality is available in Basic MSI projects.

Note that if you want to add the print or launch support to your project but your installation targets systems that have Windows Installer 4.5 or earlier, consider using the support that InstallShield provides. For information on adding the print support, see "Specifying Dialogs for Your Installation in the Project Assistant" in the InstallShield Help Library. For information on adding the print support, see "Adding a Print Button to a Dialog." The support that is available with InstallShield does not require Windows Installer 5.

### **Windows Installer 4.5 Prerequisites for Windows Vista Systems**

The conditions in the following two InstallShield prerequisites are now configured correctly: Windows Installer 4.5 for Windows Vista (x86) and Windows Installer 4.5 for Windows Vista (x64). If one of these InstallShield prerequisites is installed on the target system during your installation, the installation no longer attempts to install it again.

### **Dependency Scanning and COM Extraction Now Exclude Protected Windows 7 Files and Registry Entries**

The Filters.xml file that is installed with InstallShield has been updated with exclusions for files and registry keys that are protected by Windows Resource Protection (WRP) on Windows 7 systems. Therefore, if you extract COM data from COM servers, InstallShield does not extract any registry keys that are protected by WRP for Windows 7. In addition, if you run the dependency scanners in InstallShield, they do not list as dependencies any files that are protected by WRP for Windows 7.

### **IOA-000049763 (Merge Module)**

The validation information for the Property table is now generated correctly in Merge Module projects that include no properties of their own. Previously in this scenario, the Value column was incorrectly marked as nullable.

### **IOA-000049954**

If you password-protect your Setup.exe file through the Setup.exe tab in the Releases view and your installation includes feature prerequisites, the installation displays a password prompt only when the end user first launches Setup.exe. The installation no longer displays a prompt when the feature prerequisites are launched.

### **IOA-000050574**

Message boxes that are launched by the InstallScript function SdShowMsg are now displayed in front of the installer window that launched SdShowMsg, but not in front of any subsequently launched windows that are opened by end users. Previously, SdShowMsg message boxes were displayed in front of all windows. This applies to InstallScript installations, InstallScript MSI installations, and Basic MSI installations that contain InstallScript custom actions and are run with a full user interface.

Note that if a function such as SdShowMsg, MessageBox, or AskYesNo is used in a custom action in a Basic MSI project and the installation is run with a limited user interface (that is, with a basic UI, a reduced UI, or no UI), the message box could be displayed behind any other open windows. This occurs because the parent window of the message box cannot be found, so the desktop is considered to be the parent window. Displaying any UI from any type of custom action that is sequenced in the Execute sequence is strongly discouraged; if this is avoided, the issue with SdShowMsg, MessageBox, and other functions displaying behind other windows does not occur.

### **IOA-000050575**

If you are creating an operating system condition for an InstallShield prerequisite, you can now specify the number zero for the service pack maximum; this indicates that no service pack should be installed on the target system in order for the condition to evaluate as true. Previously if you entered the number zero, the installation behaved as though no service pack maximum was specified for the InstallShield prerequisite.

### **IOA-000050816 (Transform)**

The Specify Output File Name panel in the Transform Wizard now lets you specify the location and name of the file by using the Browse button. Previously, if you used the Browse button to specify the location and name, a "file not found" error was displayed; to work around the issue, it was necessary to manually enter the location and name.

### **IOA-000050944 (InstallScript)**

If you use a custom dialog skin in a project that contains setup prerequisites, the installation no longer displays a blank area on the setup prerequisite dialog at run time. Previously, the list of setup prerequisites disappeared from this dialog after end users selected a prerequisite in the list and then clicked the Install button.

### **IOA-000051224 (InstallScript, InstallScript MSI)**

If the InstallScript functions PlaceBitmap and SetTitle are used to show a full-screen bitmap image and specify the text on the image, the specified title is now displayed correctly on the image. Previously, part of the title was cut off.

### **IOA-000051686 (Virtualization)**

When InstallShield is determining the target folders that are represented by each directory in the .msi package during the App-V conversion process, it checks to see if the package contains set-a-property and set-a-directory custom actions that alter the directory target location. Now InstallShield has better handling for this scenario, enabling the conversion process to better resolve Windows Installer variables that are used by the custom action to set the directory target; this results in fewer -9137 conversion warnings. Previously, the conversion process failed to determine the correct directory target location when such a custom action used properties or references to other Windows Installer variables such as environment variables to specify the target.

### **IOA-000051784 (Basic MSI, InstallScript MSI)**

When you have specified release flags and product configuration flags for a release that you are building in the Releases view, and the release includes an InstallShield prerequisite that also has a release flag that needs to be included in the release, the installation installs the prerequisite if appropriate. Previously, if no prerequisites were included without such flags, the installation ignored the prerequisite. In addition, if any feature prerequisites were included, the installation would relaunch itself after the end user clicked the Install button.

### **IOB-000058268 (Virtualization)**

When creating a virtual package, InstallShield can now decompress certain .msi packages that could not previously be decompressed. Previously in certain cases, virtual conversion error -9006 occurred for these .msi packages.

### **IOC-000081344**

If a file is selected on the Application to Run tab in the InstallShield Prerequisite Editor, that file is the one that is launched on the target system at run time when the installation runs that InstallShield prerequisite. Previously, if the InstallShield prerequisite also included a file that started with the same string of characters as the file that was selected to be run, the installation may have launched the wrong file.

### **IOC-000081558 (Basic MSI)**

If you are using the Unicode setup launcher type and your project includes one or more Unicode-only languages (such as Hindi), the installation can now be run in those languages. Previously when end users launched the

Setup.exe file, run-time error 1208 occurred; the error message stated that the code page for the specific language was not available on the target system.

#### **IOC-000081594 (Virtualization)**

If you use the App-V Application Launcher utility (AppVLauncher.exe) to launch an App-V application that does not contain any shortcuts, the App-V Application Launcher now displays a more informative message: "No application OSD files were found. Most App-V packages have OSD files which are entry points (similar to shortcuts) to launch applications." The previous message was, "There are no application OSD files."

#### **IOC-000081717 (InstallScript, InstallScript MSI)**

If you upgrade a project from InstallShield 2009 or earlier to InstallShield 2010 SP1, and the earlier project contained one or more dialogs that were edited on a machine that had a large or custom DPI setting selected, the dialogs are now imported into the project correctly. Previously, the dialogs were smaller than they should have been.

#### **IOC-000081969 (Basic MSI, InstallScript MSI)**

If a multilanguage installation is run on a 64-bit target system and the installation contains an InstallScript custom action that has Deferred Execution in System Context selected for its In-Script Execution setting, that custom action can now be executed successfully. In addition, Windows Installer no longer crashes in that scenario.

#### **IOC-000082018 (Basic MSI, InstallScript, InstallScript MSI)**

Some IIS settings that are configured in the Internet Information Services view and that were previously configured incorrectly at run time are now configured correctly at run time.

In addition, if your project includes an application pool that you configured through the Internet Information Services view, you select the SpecificUser option for the application pool's Identity setting, and you specify a password in the SpecificUser Password setting, the password is now set correctly at run time.

#### **IOC-000082019 (Basic MSI, InstallScript MSI)**

If a project that was created in an earlier version of InstallShield contains a Windows Mobile installation, and that project is upgraded to InstallShield 2010 SP1, the Windows Mobile application is now installed at run time. Previously at run time, the Application Manager (CeAppMgr.exe) was not launched, so the Windows Mobile application could not be installed.

#### **IOC-000082066 (Basic MSI)**

InstallShield now includes a new 64-bit EulaScrollWatcher.dll file that you can use in a custom action that will run on 64-bit target systems. This DLL enables you to require end users to scroll through the EULA in the LicenseAgreement dialog. Previously, the only EulaScrollWatcher.dll file that InstallShield included would fail on 64-bit systems.

For detailed instructions, see "Requiring End Users to Scroll Through the EULA in the LicenseAgreement Dialog" in the InstallShield Help Library.

#### **IOC-000082210 (Virtualization)**

When you build an App-V package, the Output window now displays a link to the build output folder for the virtual package.

#### **IOC-000082370 (Virtualization)**

The Package Information page of the Microsoft App-V Assistant now includes options for targeting Windows 7 and Windows Server 2008 R2.

### **IOC-000082427 (Basic MSI, InstallScript MSI)**

Build error -1530 no longer occurs if you try to build a network image release that uses compression and automatic disk spanning, and if the release includes a large file for an InstallShield prerequisite whose location is Copy from Source Media.

### **IOC-000082470 (Virtualization)**

The Microsoft App-V Assistant now lets you add shortcuts that point to files that are not in the App-V package.

### **IOC-000082496 (Virtualization)**

The App-V Application Launcher utility (AppVLauncher.exe) now adds the App-V package to the client and uses the OVERRIDEURL option, which overrides the application source root (ASR) setting. As a result of adding the package instead of just individual OSD applications, the file extensions and shortcuts of the package are now published to the machine. Previously, the AppVLauncher.exe failed to successfully launch OSD files in an App-V client environment that had the ASR set.

### **IOC-000082545**

When InstallShield is used within Visual Studio and a referenced project has a short name, InstallShield now resolves project output groups correctly, and build error -6221 no longer occurs.

### **IOC-000082568 (Virtualization)**

In OSD files, paths for items such as shortcut parameters and environment variables now correctly refer to the root drive of the machine by using C:\ instead of %drive\_c%\.

### **IOC-000082570 (Virtualization)**

If you choose to include an installation package for the App-V package as part of the App-V package build output and you specify to have the application loaded from the shared location (thus, the .sft file is not included in the installation package), the installation package now determines whether the .sft file is in the same folder as the installation package. If the .sft file is in the same folder, the .sft file location is now passed to the App-V client when the application is loaded. If the .sft file is not in the same folder, the location that is specified in the manifest file and the OSD files is used. Previously, the location that was specified in the manifest file and the OSD files was used regardless of whether the .sft file was in the same folder.

### **IOC-000082597 (Basic MSI)**

When Setup.exe launches a 64-bit .msi package, it now launches the 64-bit version of Windows Installer to work around a Windows Installer problem. As a result, a 64-bit minor upgrade (or a 64-bit installation run in maintenance mode) that updates files in System64folder no longer generates run-time error -2324.

### **IOC-000082647 (Basic MSI, InstallScript MSI, QuickPatch)**

If you create an installation that uses the MsiServiceConfig, MsiServiceConfigFailureActions, or MsiLockPermissionsEx tables—which are new for Windows Installer 5—the patch cannot be marked as uninstallable. Thus, Val0015 now displays a warning if any of these tables are included in a release when upgrade and patch validation is performed.

### **IOC-000082648 (Virtualization)**

InstallShield includes a new machine-wide setting that lets you specify an existing writable location where all virtual packages should be built. You may want to use this global setting if you are converting .msi packages that are in a read-only location to virtual packages.

To configure the machine-wide setting, find the Settings.xml file that is installed with InstallShield. In that Settings.xml file, enter the path as the text content for the <GlobalBuildRedirectFolder> element, which is a child element of the <Virtualization> element.

### **IOC-000082651**

The IIS scanner (IISscan.exe) now scans non-default Web sites and creates XML files that record the values of the settings that you can configure in the Internet Information Services view in InstallShield. Previously in this scenario, the IIS scanner displayed an error message that indicated that the Web site could not be found; in addition, the XML file was not generated.

### **IOC-000082757 (Basic MSI)**

InstallShield no longer adds the SQLBrowse dialog to your project if you upgrade it from an earlier version of InstallShield to InstallShield 2010 SP1, and if the project does not contain a SQL connection that was configured in the SQL Scripts view.

### **IOC-000083046 (Virtualization)**

InstallShield now creates the .osd, manifest, and .sprj files with UTF-8 encoding when building App-V packages; this matches the behavior of the Microsoft Application Virtualization Sequencer. Previously, InstallShield created these files with UTF-16 encoding, which seemed to lead to parser errors on some systems that had Cumulative Update 1 for Microsoft Application Virtualization 4.5.

## **New in InstallShield 2010 Original Release Version (June 18, 2009)**

Note: Without SP1, InstallShield 2010 supports Microsoft Windows 7 Release Candidate English, Microsoft Windows Server 2008 R2 Release Candidate English, and the beta version of Windows Installer 5.

## **Features in InstallShield 2010 Original Release Version**

### **Support for Creating Customized Virtual Applications**

Now you can use InstallShield to create customized virtual applications in the Microsoft App-V format. Virtualization enables you to isolate an application in its own environment so that it does not conflict with existing applications or modify the underlying operating system.

Virtual applications run in virtual environments that keep the application layer and the operating system layer separate. Each application includes its own configuration information in its virtual environment. As a result, many applications can run side-by-side with other applications on the same computer without any conflicts.

To create a virtual application, use the new Microsoft App-V tab that is available in the following project types: Basic MSI and MSI Database.

The virtualization support is available if you purchase InstallShield with the Virtualization Pack.

InstallShield also includes InstallShield prerequisites for the Microsoft App-V 4.5 Desktop Client installation and the Microsoft Application Error Reporting installation. The Application Error Reporting prerequisite is a dependency of the App-V prerequisite. The redistributable files for these InstallShield prerequisites are not available for download from within InstallShield, since you must obtain them from Microsoft. Once you obtain them from Microsoft, place them in the location that is displayed when you are editing these prerequisites in the InstallShield Prerequisite Editor.

### **Ability to Target Windows 7 and Windows Server 2008 R2 Systems**

InstallShield enables you to specify that your installation requires Windows 7 or Windows Server 2008 R2. It also lets you build feature and component conditions for these operating systems.

### **Windows 7 and Windows Server 2008 R2 Support for Displaying Installation Progress on the Taskbar**

Installations that are run on Windows 7 and Windows Server 2008 R2 now show a progress bar on the Windows taskbar during file transfer. This applies to InstallScript and InstallScript MSI installations. In addition, it applies to Basic MSI installations that display billboards that were configured in the Billboards view. Note that a progress bar is not displayed on the taskbar on earlier versions of Windows. It is also not displayed during setup initialization or while InstallShield prerequisites are being installed.

### **Beta Windows Installer 5 Support for Per-User Installations**

The General Information view has a new Show Per-User Option setting. This setting lets you specify whether you want the ReadyToInstall dialog—in certain scenarios—to include buttons that let end users indicate how they want to install the product: for the current user or for all users. The per-user button sets the new Windows Installer property MSIINSTALLPERUSER equal to 1 to indicate that the package should be installed for the current user. The MSIINSTALLPERUSER property is available with the beta of Windows Installer 5.

This feature is available in Basic MSI projects.

### **Beta Windows Installer 5 Support for Reducing the Time for Installing Large Packages**

Use the new Fast Install setting in the General Information view to select one or more options that may help reduce the time that is required to install a large Windows Installer package. For example, you can specify that you do not want a system restore point to be saved for your installation. You can also specify that you want the installation to perform file costing, but not any other costing.

This setting configures the new Windows Installer property MSIFASTINSTALL, which can be set at the command line. Windows Installer 5 includes support for this property. Earlier versions of Windows Installer ignore it.

This setting is available in the following project types: Basic MSI, InstallScript MSI, MSI Database, and Transform.

### **Beta Support for Additional Windows Installer 5 Features**

The Director Editor in InstallShield includes beta support for the new Windows Installer 5 tables (MsiLockPermissionEx, MsiServiceConfig, MsiServiceConfigFailureActions, and MsiShortcutProperty).

This feature is available in the following project types: Basic MSI and InstallScript MSI.

### **Ability to Detect the Presence of a Virtual Machine**

InstallShield lets you determine whether an installation is running on any of the following types of virtual machines:

- Microsoft Hyper-V
- A VMware product such as VMware Player, VMware Workstation, or VMware Server
- Microsoft Virtual PC

Two new Windows Installer properties are available when you add an MSI DLL custom action to your project to detect virtual machines: IS\_VM\_DETECTED and IS\_VM\_TYPE. The custom action should be configured to call the ISDetectVM function in the SetAllUsers.dll file, which is installed with InstallShield.

In addition, the InstallScript language has been expanded to support the detection. The structure SYSINFO contains a new bIsVirtualMachine member, and a new VIRTUAL\_MACHINE\_TYPE constant is available for use with the InstallScript function GetSystemInfo.

This feature is available in the following project types: Basic MSI, InstallScript, InstallScript MSI, InstallScript Object, and Merge Module.

For more information, see "Detecting Whether the Installation Is Being Run on a Virtual Machine" in the InstallShield Help Library.

### Support for 64-Bit COM Extraction

InstallShield now supports 64-bit COM extraction. If you are using InstallShield on a 64-bit operating system, InstallShield can extract COM data from a 64-bit COM server. In order to install the data to the correct locations, the component must be marked as 64 bit.

InstallShield must be installed on a 64-bit operating system in order to perform 64-bit COM extraction.

This feature is available in the following project types: Basic MSI, InstallScript MSI, and Merge Module.

This feature resolves issue IOB-000005298.

### New Support for Setting Permissions for Files, Folders, and Registry Keys

InstallShield offers two new ways to secure files, folders, and registry keys for end users who run your product in a locked-down environment:

- **Custom InstallShield handling**—In Windows Installer–based projects, you can choose to use custom support for setting permissions at run time. With this option, InstallShield stores permission information for your product in the custom ISLockPermissions table of the .msi database. InstallShield also adds custom actions to your project to set the permissions. This support is available in the following project types: Basic MSI, InstallScript MSI, Merge Module, MSI Database, MSM Database, and Transform.
- **SetObjectPermissions, an InstallScript Function**—You can use the new SetObjectPermissions function in InstallScript events and InstallScript custom actions to set permissions at run time. You can use this function in the following project types: InstallScript, Basic MSI, InstallScript MSI, and Merge Module.

With the custom InstallShield handling option, the file, folder, or registry key for which you are setting permissions must be installed as part of your installation. With the SetObjectPermissions function, the file, folder, or registry key can be installed as part of your installation, or it can be already present on the target system.

Previously, the only option that InstallShield offered for setting permissions was to use the traditional Windows Installer handling. With this option, the permission information is stored in the LockPermissions table of the .msi database. The new custom InstallShield handling option and the SetObjectPermissions function offer several advantages over the traditional Windows Installer handling:

- The custom option and the SetObjectPermissions function include support for many well-known security identifiers (SIDs) that are not supported by the traditional Windows Installer handling option.
- The custom option and the SetObjectPermissions function support the use of localized user names for the supported SIDs, unlike the traditional option. With the traditional option, if you try to use a localized name to set permissions on a non-English system, the installation may fail.
- The custom option and the SetObjectPermissions function let you specify that you want to deny a user or group from having the permissions that you are specifying. The traditional handling does not allow you to do this.
- The custom option and the SetObjectPermissions function let you add permissions to a file, folder, or registry key that already exists on the target system, without deleting any existing permissions for that object. With the traditional handling, the existing permissions are deleted.
- The custom option and the SetObjectPermissions function let you configure permissions for a folder (or a registry key), and indicate whether you want the permissions to be applied to all of the folder's subfolders and

files (or the registry key's subkeys). With the traditional handling, if you want to configure permissions for a subfolder or a file in a folder (or a subkey under a registry key), the parent that is created on the target system automatically inherits the permissions of its child.

- The custom option and the SetObjectPermissions function let you configure permissions for a new user that is being created during the installation. The traditional handling does not allow you to do this; the user must already exist on the target system at run time.

The General Information view has a new Locked-Down Permissions setting that lets you specify whether you want to use the new custom InstallShield handling or the traditional Windows Installer handling for all new permissions that you set for files, folders, and registry keys in your project. If you have already configured some permissions in your project and then you change the value of this setting, InstallShield lets you specify whether you want to use the alternate handling method for those already-existing permissions. In all new projects, the default value for this setting is the custom InstallShield handling option. If you upgrade a project from InstallShield 2009 or earlier to InstallShield 2010, the traditional Windows Installer handling option is the default value of this setting. This new setting is available in the following project types: Basic MSI, InstallScript MSI, Merge Module, MSI Database, MSM Database, and Transform.

This feature resolves the following issues: 1-784A7, 1-JOMKL, IOA-000030286, IOA-000032834, IOA-000044784, IOC-000044382, IOC-000062641.

### Support for InstallShield Prerequisites in InstallScript Projects

InstallShield now enables you to add InstallShield prerequisites to InstallScript projects. Previously, only Basic MSI and InstallScript MSI projects included support for this type of redistributable.

InstallShield prerequisites are redistributables that usually install a product or technology framework that is required by your product. Now you can add any of the InstallShield prerequisites that are provided with InstallShield—or any of your own custom-designed InstallShield prerequisites—to InstallScript projects. If you work on a mix of different project types, InstallShield lets you simplify your testing matrix by enabling you to reuse this type of redistributable in all of your Basic MSI, InstallScript, and InstallScript MSI projects.

To add an InstallShield prerequisite to an InstallScript project, use the new Prerequisites view. InstallScript projects include support for the setup prerequisite type of InstallShield prerequisite, which is run before the main installation's user interface is run. InstallScript projects do not have support for feature prerequisites, which are InstallShield prerequisites that are associated with features.

### New InstallShield Prerequisites for Windows Installer, .NET Framework, Crystal Reports, and Other Redistributables

InstallShield includes a number of new InstallShield prerequisites that you can add to Basic MSI, InstallScript, and InstallScript MSI projects:

- Windows Installer 4.5 (The InstallShield prerequisites for Windows Installer 4.5 include the fix that is described in Microsoft KB958655.)
- Windows Installer 4.5 Update (The InstallShield prerequisites for the Windows Installer 4.5 Update include the fix that is described in Microsoft KB958655. Windows Installer 4.5 must already be installed on the target system for this update.)
- Windows Installer 3.1, Windows Installer 3.0, and Windows Installer 2.0 (These versions of Windows Installer redistributables were previously available if you added Windows Installer to your project in the Releases view. These versions were not previously available as InstallShield prerequisites.)
- .NET Framework 3.0 SP1
- .NET Framework 2.0 SP2

- Internet Explorer 8
- Microsoft SQL Server 2008 Express SP1
- Microsoft SQL Server 2005 Express SP3
- Microsoft Visual C++ 2005 SP1 Redistributable Package
- Oracle 11g Instant Client 11.1.0.7 (Oracle does not provide an installer for the Oracle Instant Client files, so you need to create an .msi package before you can use this InstallShield prerequisite in your projects. You can do so easily by using the Oracle Instant Client installation project that is installed with InstallShield (*InstallShield Program Files\Folder\Support\Oracle Instant Client*.)
- Crystal Reports Basic for Visual Studio 2008 (Use this prerequisite with the Crystal Reports Basic installation that is installed with Visual Studio 2008. Note that you may need to change the path of the Crystal Reports Basic installation in the .prq file, depending on where the .msi package is located on your system.)

This feature resolves the following issues: IOA-000047620, IOA-000047761, IOC-000078966, IOC-000080736.

### Microsoft SQL Server 2008 SP1 Support

InstallShield now includes support for running SQL script on SQL Server 2008 SP1. In addition, InstallShield includes SQL Server 2008 SP1 in the predefined list of database servers that you can select when you are specifying in the SQL Scripts view the target database servers that your product supports.

### Oracle 11g Support

InstallShield now includes support for running SQL script on Oracle 11g. In addition, InstallShield includes Oracle 11g in the predefined list of database servers that you can select when you are specifying in the SQL Scripts view the target database servers that your product supports.

This feature resolves issue IOC-000080690.

### SQL Server Compact Edition 3.5 SP1 Support Now Available for Mobile Devices

InstallShield now includes support for SQL Server Compact Edition 3.5 SP1. If you have SQL Server Compact Edition 3.5 SP1 installed on your build machine and you select the SQL Server Compact 3.5, SQL Server Compact 3.5 Replication, or SQL 3.5 Client options in the Windows Mobile Wizard or the Smart Device Setup Wizard, InstallShield includes .cab files of SQL Server Compact Edition 3.5 SP1 in your installation at build time. Note that you need to obtain the SQL Server Compact Edition 3.5 SP1 redistributable from the MSDN Web site and install it on your build machine.

This feature applies to the following project types: Basic MSI, InstallScript MSI, and Smart Device.

### New Tool for Scanning an IIS Web Site, Recording Its Settings, and Importing Those Settings into an InstallShield Project

InstallShield includes an IIS scanner (IISscan.exe), a new command-line tool for scanning an existing IIS Web site and recording IIS data about the Web site. The IIS scanner creates an XML file that contains all of the settings for the Web site, its virtual directories, its applications, and its application pools. You can use the XML file to import the IIS data into the Internet Information Services view in InstallShield. Once you have imported the IIS data into a project, you can use the Internet Information Services view to make changes to the IIS settings as needed.

The ability to import the IIS data into an InstallShield project is available only in the Premier edition of InstallShield.

### Ability to Add IIS Web Applications to Web Sites, Plus Other IIS-Related Improvements

InstallShield now lets you add IIS Web applications to Web sites. You can do so by right-clicking a Web site in the Internet Information Services view and clicking New Application. Once you have added a new application, you can configure its settings in the right pane.

InstallShield also lets you create a virtual directory without an application. Previously whenever you created a virtual directory, an application was also created automatically.

In addition, InstallShield has new IIS settings:

- **Managed Pipeline Mode**—This setting enables you to specify the appropriate request-processing pipeline mode—either integrated or classic—for an application pool.
- **Enable 32-Bit Applications**—This setting lets you specify whether you want to allow 32-bit applications in the selected application pool to be run on 64-bit systems.
- **.NET Framework Version**—This setting is where you specify the version of the .NET Framework that an application pool should load.
- **ASP.NET Platform**—If your installation may be run on a 64-bit version of Windows with the .NET Framework, use this setting to specify which ASP.NET platform should be used to map a Web site, application, or virtual directory to the ASP.NET version.

This feature is available in the following project types: Basic MSI, InstallScript, and InstallScript MSI.

This feature resolves the following issues: 1-PLEIX, IOA-000045314, IOC-000057470, IOC-000059119, IOC-000065672, IOC-000066474, IOC-000067040, IOC-000069568, IOC-000071377, IOC-000072349, IOC-000073995, IOC-000079857, IOC-000079976.

## New View for Configuring Run-Time Text File Changes

InstallShield has a new Text File Changes view, which enables you to configure search-and-replace behavior for content in text files—for example, .txt, .htm, .xml, .config, .ini, and .sql files—that you want to modify at run time on the target system. The text files can be part of your installation, or they can be files that are already present on target systems.

You can use Windows Installer properties to specify the names of the text files that you want to include in or exclude from your search. You can also use properties to specify the search strings and the replacement strings. This enables you to use data that end users enter in dialogs, or other configuration information that is determined at run time, when your product's text files are modified on the target system. For example, if your project includes a dialog in which end users must specify an IP address, your installation can search a set of files for a token, and replace it with the IP address that end users enter.

Note that the Text File Changes view offers an alternative for configuring XML file changes in the XML File Changes view. Using the Text File Changes view offers some advantages. For example, this new view does not have any run-time requirements; however, changes that are configured through the XML File Changes view require MSXML to be present on the target system. In addition, configuring changes in the Text File Changes view does not require you to enter XPath queries, as is required in the XML File Changes view.

The Text File Changes view is available in the following project types: Basic MSI and InstallScript MSI.

## New String Editor View

InstallShield has a new String Editor view. This view contains a spreadsheetlike table that shows the collection of language-independent identifiers and corresponding language-specific values for your project. In the String Editor view, you have complete and centralized control over the localizable text strings that are displayed at run time during the installation process. The view replaces the string tables that were previously available as nodes within the General Information view. Following is a list of some of the highlights of the new view:

- The view has a toolbar with buttons that let you add, edit, delete, find, replace, export, and import string entries. It also has a button that lets you search the project to identify all of the instances in which a specific string identifier is used.

- The top of the view has a new group box area; you can drag and drop column headings onto this area to organize the rows in the view in a hierarchical format. This functionality makes it easy to sort string entries by categories such as language and by modified date.
- This view has support for dynamic searches—as you are typing a string in the search box, InstallShield hides all of the string entries that do not contain it.

This feature is available in the following project types: Basic MSI, InstallScript, InstallScript MSI, InstallScript Object, Merge Module.

This feature resolves issue 1-EJNEY.

## Support for Unicode

InstallShield takes a three-pronged approach to fully supporting modern multilanguage installations: Windows Installer databases can now be built in a Unicode format, InstallShield projects are now stored in a Unicode format, and the InstallShield interface now supports entering and viewing Unicode characters from multiple character sets at the same time.

### Unicode (UTF-8) Databases

The Build tab in the Releases view has a new Build UTF-8 Database setting that lets you specify that a Windows Installer database, along with any instance or language transforms, be built using the UTF-8 encoding. The UTF-8 encoding supports characters from all languages simultaneously, enabling you to mix and match, for example, Japanese and German, or Russian and Polish, both in text shown to end users and in file names and registry keys. These mixed languages work correctly regardless of the current language of the target system. Unicode support has also been added to key parts of the installation run times, including IIS support and changes to text and XML files.

The default value for the new Build UTF-8 Database setting is No.

The automation interface now includes support for this new setting. The ISWiRelease object includes a new BuildUTF8Database property that lets you specify whether you want to use the UTF-8 encoding.

This feature is available in the following project types: Basic MSI, InstallScript MSI, and Merge Module projects.

This feature resolves the following issues: IOB-000050571, IOC-000053626, IOC-000070145, and IOC-000074276.

### Unicode Project Files (\*.ism)

InstallShield now uses the UTF-8 encoding when saving both binary and XML project files. Because of this change, the files, registry data, and other strings that are used in the project can include characters from all languages simultaneously. With this encoding, InstallShield no longer needs to use an unreadable Base64 encoding for strings that are stored in the ISString table. Instead, as you add or modify strings in a project, InstallShield now saves them as human-readable Unicode strings that you can easily examine for changes across revisions of your project. Therefore, InstallShield uses only Unicode strings for all new projects that are created in InstallShield 2010; for upgraded projects, InstallShield uses Unicode for new and modified strings, as well as for strings that have been exported and reimported.

If you use Unicode values that cannot be represented in the target build (for example, an InstallScript installation, or a Basic MSI installation in which No is selected for the Build UTF-8 Database setting), a new build error points to the item that needs to be changed. In some instances, this reveals invalid string entries that were silently allowed in earlier versions of InstallShield.

This feature applies to all project types.

### Unicode Views in InstallShield

Many views in InstallShield have been enhanced to display and allow you to enter characters from all languages.

For example, in the Files and Folders view, Chinese file names from your local English system are no longer displayed with question marks for their names, and now you can add these files to your project. Similarly, the Registry view no longer converts Thai registry keys or values to question marks, and you can install them with your Windows Installer–based projects. In addition, you can view and edit strings from all languages in the String Editor view, a new separate view; previously, string entries were available from separate language nodes in the General Information view. Examples of other enhanced views include the Property Manager, Path Variables, Direct Editor, General Information, and Setup Design views.

Note that whenever No is selected for the Build UTF-8 Database setting, all file names, registry keys, and other strings must still consist of characters from the code page of all target languages that will use it. In this scenario, if an item uses a character that is not available on the code page of a target language, InstallShield reports a new build error that points to this item; the Chinese file name cannot be used in an English installation unless Yes is selected for the Build UTF-8 Database setting.

This feature applies to all project types.

This feature resolves the following issues: 1-12AZWL, 1-17D9Y0, 1-AT26Z, 1-NMATB, 1-SORS9, IOC-000073872.

### **New Billboard Support in Basic MSI Projects**

Basic MSI projects now include support for billboards. You can add billboards to your projects to display information to end users during the installation process. The billboards can be used to communicate, advertise, educate, and entertain end users. For example, billboards can present overviews on new features of the product being installed or other products from your company. Each billboard is a file that you or your company's graphics department creates for complete control over the look and feel of the file transfer.

Following are some of the highlights regarding billboard support in Basic MSI projects:

- You can add an Adobe Flash application file (.swf) as a billboard in your project. Flash application files can consist of videos, movies, sounds, interactive interfaces, games, text, and more—anything that is supported by the .swf type of file.
- InstallShield lets you use .bmp, .gif, .jpg, and .jpeg files as billboards.
- InstallShield includes a Billboard Type setting that lets you specify which style of billboard you want to use in your installation. For example, with one style, the installation displays a full-screen background, with billboards in the foreground, and a small progress box in the lower-right corner of the screen. With another style, the installation displays a standard-size dialog that shows the billboards. The bottom of this dialog shows the progress bar.
- InstallShield lets you preview a billboard to see how it would be displayed at run time, without requiring you to build and run a release. Previewing a billboard lets you see how your billboard will look with the background color, position, and related settings that are currently configured for your billboard.

The Billboards view in InstallShield is where you add billboard files, configure billboard-related settings, and preview billboards.

Previously, only InstallScript and InstallScript MSI projects included support for billboards. Note that billboard support in these project types is different than support in Basic MSI projects.

### **Expanded Billboard Support in InstallScript and InstallScript MSI Projects**

InstallScript and InstallScript MSI projects now let you use .gif, .jpg, and .jpeg files as billboards. Previously, only .bmp files were supported. Use the Support Files/Billboards view to add billboards to your project.

Billboard files in InstallScript and InstallScript MSI projects must follow a designated naming convention. Each file name must begin with bbrd, followed by the number of the billboard (from 1 through 99); each must end with a

supported file extension (.bmp, .gif, .jpg, or .jpeg). Note that it is no longer necessary for the file name numbers to be contiguous; that is, you can add bbrd1.jpg, bbrd3.jpg, and bbrd5.jpg to your project, and each image is displayed at run time in order. Previously, there could not be any numbers missing from the file name sequence for your billboards. For example, if you added bbrd1.bmp, bbrd2.bmp, and bbrd4.bmp to your installation project, then bbrd1.bmp and bbrd2.bmp were displayed at run time; however, bbrd4.bmp was not displayed, since bbrd3.bmp was missing from the sequence.

### **Ability to Play an Adobe Flash Application File (.swf) with the InstallScript Function PlayMMedia**

The InstallScript function PlayMMedia now includes support for playing an Adobe Flash application file (.swf) on a background window during InstallScript and InstallScript MSI installations. Flash application files can consist of videos, movies, sounds, interactive interfaces, games, text, and more—anything that is supported by the .swf type of file.

If you are using a Flash file, you can use SizeWindow and PlaceWindow to control the size and placement of the background window that displays the Flash file.

### **Support for HTML Controls on Dialogs During InstallScript and InstallScript MSI Installations**

InstallShield includes support for HTML controls on dialogs in InstallScript and InstallScript MSI projects. HTML controls enable you to use HTML markup for dialog controls. You can include on dialogs links to Web pages, installed HTML files, and HTML support files. If an end user clicks the hyperlink on the run-time dialog, you can have the HTML page open in an Internet browser, or you can trigger other behavior that you have defined through your InstallScript code. The HTML control lets you use any valid HTML markup, including styles to control their appearance.

The HTML control also lets you display the HTML content directly on a dialog if the content is an installed HTML file or HTML support file.

A new InstallScript function called CtrlGetUrlForLinkClicked is available. This function retrieves the URL for a link that an end user clicked.

### **Ability to Add Unsupported Languages to InstallScript Projects**

The InstallShield Premier edition includes default run-time strings for InstallScript projects in 33 different languages. The InstallShield Premier edition also lets you use the New Language Wizard to add unsupported languages, beyond the built-in 33 languages, to InstallScript projects. Once you have added unsupported languages to an InstallScript project, you can use the String Editor view to provide translated strings for the unsupported languages. As an alternative, you can export a language's string entries to a file, translate the string values in the file, and then import the translated file into your project. Previously, the New Language Wizard enabled you to add unsupported languages to only Basic MSI and InstallScript MSI projects.

This feature resolves issue IOC-000068232.

### **Ability to Specify Paths for Libraries to Which the InstallScript Compiler Should Link**

The Compile/Link tab on the Settings dialog box has a new Additional Library Paths box that lets you specify the locations where the InstallScript compiler should search for InstallScript libraries (.obl files) that should be linked to your script. In addition, when you specify your custom libraries on this tab, you can now specify just the file name, without the full path. These changes enable you to move your libraries to different directories but still successfully compile your script without manually changing the path for libraries on the Compile/Link tab.

This feature resolves issue 1-4X5BD.

## New FLEXnet Connect 11.6 Redistributables Available

InstallShield includes support for FLEXnet Connect 11.6 in Basic MSI and InstallScript MSI projects. Use the Update Notifications view in InstallShield to include one of the two FLEXnet Connect 11.6 merge modules—one has the Common Software Manager, and the other does not. These merge modules replace the FLEXnet Connect 11 merge modules. For details about the changes in the updated merge modules, see the [FLEXnet Connect release notes \(PDF\)](#).

## DIFx 2.1.1 Support

InstallShield includes support for the latest version of Driver Install Frameworks for Applications (DIFx 2.1.1). This new version, which includes the latest binary files from Microsoft, is available for the following project types: Basic MSI, InstallScript, and InstallScript MSI.

This feature resolves issue IOC-000080214.

## Enhancements in InstallShield 2010 Original Release Version

### Usability Enhancements

Many of the views in InstallShield have been enhanced to improve productivity and usability. For example, several views contain new toolbars that make options easier to find. Some of the views that contain grids let you customize how the rows in a grid are organized. Searches are performed more quickly in the views that offer search capabilities. Following are examples of some of the highlights:

- **Direct Editor view**—When you select a table in this view, a new toolbar is displayed. The toolbar has buttons that let you add, cut, copy, paste, find, and replace data in the table. This view also has support for dynamic searches—as you are typing a string in the search box, InstallShield hides all of the table records that do not contain it. When you are adding or editing a record, InstallShield displays help text for each field.
- **Property Manager view**—This view contains a new toolbar that has buttons for performing tasks such as adding and deleting properties. This view also has support for dynamic searches—as you are typing a string in the search box, InstallShield hides all of the table records that do not contain it. The top of the view has a new group box area; you can drag and drop column headings onto this area to organize the rows in the view in a hierarchical format. You can now select multiple properties in this view (by using the mouse and the SHIFT or CTRL button) and then delete them all at once.
- **Redistributables view**—The new toolbar and the new group box area in this view provide robust search and organizational functionality. You can drag and drop column headings onto the group box area to organize the list of redistributables in a hierarchical format. In addition, you can type a string in the toolbar's search box, and InstallShield hides all of the redistributables that do not contain it.
- **Internet Information Services view**—This view has been redesigned to look similar to IIS 7: the settings are now displayed in grids, instead of on tabs. The grids have buttons that let you sort the grid settings by category or alphabetically. When you select a setting in one of the grids in this view, InstallShield displays help information for that setting in the lower-right pane.
- **General Information view**—All of the settings in this view are displayed in one grid, instead of as separate grids that are associated with nodes within this view. The settings are grouped into several categories to make it easy to find a particular setting. In addition, the grid has a button that lets you sort the grid settings by category or alphabetically. The string tables that were previously in this view have been moved to a new String Editor view.
- **Path Variables view**—This view contains a new toolbar that has buttons for performing tasks such as adding and deleting path variables. This view also has support for dynamic searches—as you are typing a string in the

search box, InstallShield hides all of the rows that do not contain it. The top of the view has a new group box area; you can drag and drop column headings onto this area to organize the rows in the view in a hierarchical format. You can now select multiple path variables in this view (by using the mouse and the SHIFT or CTRL button) and then delete them all at once.

In addition, the Output window, which is displayed when you are building a release, performing validation, or compiling script, has been enhanced. The Output window or its individual tabs can be docked to any side of the workspace in InstallShield, or they can be dragged to free-floating positions. If you drag the Output window or one of its tabs to the edge of the InstallShield interface, it becomes a docked window. If you drag the Output window or one of its tabs away from any of the edges of the InstallShield interface, it becomes undocked.

This enhancement resolves the following issues: 1-FVS7U, IOA-000042399.

### **Support for Specifying Action Progress Messages**

To keep end users informed, installations commonly display text on the progress dialog to describe the installation's current activity. This usually accompanies the progress bar as a means of installation status. As each standard action and custom action is encountered, a message about the action is displayed on the progress dialog. This may be especially useful for actions that take a long time to execute. The same action text is also written to the installation's log file if one is created.

The Custom Actions and Sequences view has a new Action Text area that lets you specify action descriptions and details. This is available in the following project types: Basic MSI, InstallScript MSI, MSI Database, and Transform.

### **Ability to Detect Whether a 64-Bit System Allows 32-Bit IIS Applications to Be Run**

At run time, you may need to have your installation check the Enable32bitAppOnWin64 property on systems that have IIS 6. Depending on the requirements for your product and the results of that check, you may want the installation to skip a particular component that contains a 32-bit IIS 6 application, or a 64-bit IIS application, for example, and proceed with the rest of the file transfer.

InstallShield includes a sample Windows Installer DLL file that detects how the Enable32bitAppOnWin64 property is set on a target system. You can add a custom action for this DLL to your project. If 32-bit applications are allowed, the Windows Installer property ISIIS6APPPoolsSupports32bit is set to a value of 1; if they are not allowed, this property is not set. You can use this property in conditions to prevent or trigger certain behavior. For example, you can create a launch condition if you want the installation to exit if the ISIIS6APPPoolsSupports32bit is set or not set.

This feature is available in the following project types: Basic MSI and InstallScript MSI.

For instructions on how to add the custom action to your project, see "Considerations for Supporting IIS 6 on 64-Bit Platforms" in the InstallShield Help Library.

### **Ability to Detect Whether the IIS 6 Metabase Compatibility Feature Is Installed**

At run time, you can have your installation detect if the IIS Metabase and IIS 6 Configuration Compatibility feature is installed on a target system, or if IIS 6 or earlier is installed. Depending on the requirements for your product and the results of that detection, you may want the installation to exit and display an error message.

For example, Web service extensions can be installed on systems that have IIS 6. On systems that have IIS 7, Web service extensions can be installed only if the IIS Metabase and IIS 6 Configuration Compatibility feature is installed. Thus, you might want to configure your installation to verify that IIS 6 is present or the IIS 6 compatibility feature is installed; if either of those conditions are false, the installation would exit and display an error message.

InstallShield includes a sample Windows Installer DLL file that detects whether the IIS Metabase and IIS 6 Configuration Compatibility feature is installed. You can add a custom action for this DLL to your project. If the IIS Metabase and IIS 6 Configuration Compatibility feature is installed, the Windows Installer property `ISIISMETABASECOMPATPRESENT` is set to a value of 1; if it is not installed, this property is not set. You can use this property in conditions to prevent or trigger certain behavior. For example, you can create a launch condition if you want the installation to exit if the `ISIISMETABASECOMPATPRESENT` is set or not set.

This feature is available in the following project types: Basic MSI and InstallScript MSI.

For instructions on how to add the custom action to your project, see "Determining If a Target System Has IIS 6 or Earlier or the IIS 6 Metabase Compatibility Feature" in the InstallShield Help Library.

### **Ability to Associate a Namespace Prefix with an Attribute of an XML File's Element**

InstallShield now lets you add a namespace prefix to an attribute in the XML File Changes view. To do so, you can type the prefix, followed by a colon (:), in front of the attribute name. Previously, if an attribute contained a namespace prefix, the installation failed.

This enhancement applies to the following project types: Basic MSI, InstallScript, and InstallScript MSI.

### **Ability to Specify Whether an XML File Should Be Formatted After Run-Time Changes**

The Advanced tab for an XML file in the XML File Changes view has a new "Format XML after changes" check box that lets you specify whether you want the XML file to be formatted after the run-time changes are made to the file. When formatting the file, the installation adds indentations to the file and replaces empty-element tags with start tags and end tags. This may cause problems for web.config files, so you may want to clear this check box for your project.

This enhancement applies to the following project types: Basic MSI, InstallScript, and InstallScript MSI.

This enhancement resolves issue IOA-000028596.

### **Predefined System Searches for the .NET Framework and Internet Explorer 8**

InstallShield has two new predefined system searches:

- Microsoft .NET Framework 3.5 SP1
- Internet Explorer 8

If your installation requires any of these, you can use the System Search view or the Installation Requirements page in the Project Assistant to add these system searches to your project. When end users launch your installation, Windows Installer checks the target system to see if the requirements are met; if they are not met, the installation displays the error message that is defined for the system search.

This enhancement applies to Basic MSI and InstallScript MSI projects.

This enhancement resolves issues IOA-000047761.

### **Custom Properties Are Now Listed in Condition Builder**

The Properties list on the Condition Builder dialog box now includes properties that are added in the Property Manager view. It also includes properties from searches that are created through the System Search Wizard. Therefore, when you are creating or editing a condition in InstallShield, you can now select your own custom properties without having to manually type them.

This enhancement applies to the following project types: Basic MSI, InstallScript MSI, Merge Module, and Transform.

This enhancement resolves the following issues: IOC-000040833, IOC-000067273.

### Ability to Specify a Maximum Service Pack Number and 64-Bit Locations for InstallShield Prerequisite Conditions

The Prerequisite Condition dialog box, which is displayed when you are adding or modifying a condition for an InstallShield prerequisite in the InstallShield Prerequisite Editor, has a new field that lets you specify the maximum service pack number. Previously, it was possible to specify a minimum service pack number, but not a maximum service pack number.

In addition, the Prerequisite Condition dialog box now includes the following 64-bit locations in the box that is displayed when you are specifying file-related conditions: [CommonFiles64Folder], [ProgramFiles64Folder], and [System64Folder]. You can select any of these properties if you want to use 64-bit locations in the file path. At run time, the installation checks these 64-bit locations if the target system is 64 bit. For 32-bit systems, the installation checks the 32-bit location equivalents.

This enhancement resolves issue IOC-000080090.

### Enhancements to MSBuild Support

The InstallShield task for MSBuild includes several new parameters:

- **RunMsiValidator**—Use this parameter to specify the .cab file that you want to use for validation. This parameter is exposed as the ItemGroup InstallShieldMsiValidators when the default targets file is used.
- **PatchConfiguration**—Use this parameter to specify the patch configuration that you want to build through MSBuild. This parameter is exposed as the property InstallShieldPatchConfiguration when the default targets file is used.
- **PathVariables**—Use this parameter to override the value of a path variable. This parameter is exposed as the ItemGroup InstallShieldPathVariableOverrides when the default targets file is used.
- **PreprocessorDefines**—Use this parameter to specify preprocessor definitions for compiling InstallScript. This parameter is exposed as the ItemGroup InstallShieldPreprocessorDefines when the default targets file is used.

### Automation Interface Enhancements

Several enhancements have been made to the automation interface.

The following constants are now available for use with the OSFilter member of the ISWiComponent and ISWiRelease objects in the automation interface:

- eosWin7 (33554432)—This is for Windows 7 and Windows Server 2008 R2.
- eosWinVista (16777216)—This is for Windows Vista and Windows Server 2008.
- eosWinServer2003 (8388608)

In addition, the value for the eosAll constant is now 64028880; previously, it was 5308624. The OSFilter member applies to the ISWiComponent object in InstallScript, InstallScript MSI, and InstallScript Object projects. The OSFilter member applies to the ISWiRelease object in InstallScript and InstallScript Object projects.

The list of available property values for the IsPlatformSelected property of the ISWiComponent object has been expanded. This property now has values for 32-bit, 64-bit Itanium, AMD64, and Windows Server 2003 R2. This applies to the following project types: InstallScript and InstallScript Object.

Note that some of the values for the existing constants have changed. To learn the new values, see "ISWiComponent Object" in the InstallShield Help Library.

The AddSQLScriptEx method has been added to the ISWiSQLConnection object. Use this method to add an ISSQLScriptFile entry and generate a valid name from the passed string. The method ensures that the name of the entry that is being added to the ISSQLScriptFile table is unique and less than 47 characters in length.

The read-write RunOnLogon property has been added to the ISWiSQLScript object. This property corresponds with the Run Script During Login check box on the Runtime tab for a SQL script in the SQL Scripts view.

The read-write Condition property has also been added to the ISWiSQLScript object. This property specifies the condition that is evaluated at run time to determine whether the SQL script should be run during installation or uninstallation. If the condition evaluates to true, the script is run. This property is available for the following project types: Basic MSI and InstallScript MSI.

The read-write DisplayName property has been added to the ISWiUpgradeTableEntry object. This property gets or sets the name of an upgrade entry. This is the internal name that is displayed for an upgrade item in the Upgrades view. This property is available for the following project types: Basic MSI and InstallScript MSI.

These enhancements resolve the following issues: IOC-000075015, IOC-000075114, IOC-000075810.

## Enhancements to the InstallScript Language for Operating Systems

The following structure members and predefined constants were added to the InstallScript language:

- **SYSINFO.WINNT.bWin7\_Server2008R2**—This is a new SYSINFO structure member. If the operating system is Windows 7 or Windows Server 2008 R2, this value is TRUE.
- **SYSINFO.bWinServer2003R2**—This is a new SYSINFO structure member. If the operating system is Windows Server 2003 R2, this value is TRUE. (Note that the value of SYSINFO.WINNT.bWinServer2003 is also TRUE on this operating system.)
- **ISOSL\_WIN7\_SERVER2008R2**—This is a new predefined constant that is available for use with the FeatureFilterOS function and the SYSINFO structure variable. It indicates that the target system is running Windows 7 or Windows Server 2008 R2.
- **ISOSL\_ST\_SERVER2003\_R2**—This is a new predefined constant that is available for use with the FeatureFilterOS function and the SYSINFO structure variable. It indicates that the target system is running Windows Server 2003 R2.

In addition, some items were renamed:

- The **SYSINFO.WINNT.bWinVista** structure member does not distinguish between Windows Vista or Windows Server 2008. Therefore, the new member **SYSINFO.WINNT.bWinVista\_Server2008** is now available. The old alias is still available, but the new one may be preferred for clarity in code.
- The **ISOSL\_WINVISTA** predefined constant does not distinguish between Windows Vista or Windows Server 2008. Therefore, the new constant **ISOSL\_WINVISTA\_SERVER2008** is now available. The old alias is still available, but the new one may be preferred for clarity in code.

## Ability to Easily Override InstallScript Dialog Source Code in the InstallScript View

The event category drop-down list in the InstallScript view has a new Dialog Source option. If you select this option, the event handler drop-down list shows all of the built-in InstallScript dialogs. You can select any dialog in this list to modify its code.

This functionality applies to the following project types: InstallScript, InstallScript MSI, and InstallScript Object.

## Changes to the Major and Minor Version Registry Entries for the Uninstall Key of InstallScript Installations

InstallScript installations now create VersionMajor and VersionMinor registry values in the Uninstall key; the names of these values now match the entries that are created during Basic MSI and InstallScript MSI installations. This applies to new installations that are created in InstallShield 2010, as well as installations that are upgraded from InstallShield 2009 or earlier. Previously, in InstallShield 2009 and earlier, the names of the values that InstallScript installations created were MajorVersion and MinorVersion; these are no longer created.

In order to use the new registry values, the values of the following InstallScript constants have been changed:

- REGDB\_VALUENAME\_UNINSTALL\_MAJORVERSION is now VersionMajor instead of MajorVersion.
- REGDB\_VALUENAME\_UNINSTALL\_MINORVERSION is now VersionMinor instead of MinorVersion.

When the MaintenanceStart function is called, it creates the updated value names in the registry. By default, it also deletes the old value names if they exist. If you do not want the old value names to be deleted from target systems, you can use the new REGDB\_OPTIONS option called REGDB\_OPTION\_NO\_DELETE\_OLD\_MAJMIN\_VERSION.

If REGDB\_UNINSTALL\_MAJOR\_VERSION or REGDB\_UNINSTALL\_MINOR\_VERSION is used with the RegDBGetItem function, RegDBGetItem first checks for the new value; if the new value is found, the function returns the value data from the new value. If the new value is not found, the function automatically checks for the old value; if the old value is found, the function returns the value data from the old value.

To provide backwards compatibility, the following new constants are available:

- REGDB\_UNINSTALL\_MAJOR\_VERSION\_OLD
- REGDB\_UNINSTALL\_MINOR\_VERSION\_OLD

You can specify these constants with the RegDBGetItem, RegDBSetItem, and RegDBDeleteItem functions to get, set, and delete the old values.

The following new string constants are also available:

- REGDB\_VALUENAME\_UNINSTALL\_MAJORVERSION\_OLD is defined as MajorVersion.
- REGDB\_VALUENAME\_UNINSTALL\_MINORVERSION\_OLD is defined as MinorVersion.

This enhancement resolves the following issues: 1-Q3KUU and IOC-000073672.

### **New CtrlGetDlgItem Function for Retrieving the Window Handle of a Control in a Custom InstallScript Dialog**

A new InstallScript function called CtrlGetDlgItem is now available. The CtrlGetDlgItem function retrieves the window handle of a control in a custom dialog. CtrlGetDlgItem is similar to the Windows API GetDlgItem, except that with CtrlGetDlgItem, you can specify the InstallScript dialog name instead of the dialog's window handle.

### **New InstallScript Constant for Passing a Null Pointer for an InstallScript String to an External DLL Function**

You can use the IS\_NULLSTR\_PTR variable to pass a null pointer to an external DLL function or Windows API through a parameter that has been prototyped as an InstallScript string. This functionality works for byval string, byref string, wstring, and binary data types.

### **New StrConvertSizeUnit Function for Converting an InstallScript Size Unit Constant to a Display String**

A new InstallScript function called StrConvertSizeUnit is now available. The StrConvertSizeUnit function returns the appropriate display string for the InstallScript size unit constant that is specified.

### **New StrTrim Function for Removing Leading and Trailing Spaces and Tabs from a String**

A new InstallScript function called StrTrim is now available. The StrTrim function removes the leading and trailing spaces and tabs from a string.

### **New SdLicense\* Dialog Functions to Supersede Existing SdLicense\* Dialog Functions**

Two new InstallScript dialog functions—SdLicenseEx and SdLicense2Ex—are now available. They both display a dialog that contains a license agreement in a multi-line edit field. The license agreement can be stored in a text file (.txt) or a rich text file (.rtf).

- SdLicenseEx displays a dialog that shows a question in a static text field. The end user responds by clicking the Yes or No button. SdLicenseEx supersedes SdLicense and SdLicenseRtf.
- SdLicense2Ex displays a dialog that has two radio buttons (one for accepting the terms of the license agreement, and one for not accepting them). The Next button becomes enabled when the end user clicks the appropriate button to accept the terms of the license agreement. SdLicense2Ex supersedes SdLicense2 and SdLicense2Rtf.

### **New ListFindKeyValueString Function for Searching Lists of Key-Value Pairs**

A new InstallScript function called ListFindKeyValueString is now available. The ListFindKeyValueString function searches a string or number list for a specified value. It returns a value from an additional list that corresponds with the position of the found string in the first list. This enables you to search lists of key-value pairs for a particular key and retrieve the corresponding value.

### **New InstallScript Code Examples**

The InstallShield documentation now has sample code for the following InstallScript functions:

- AdminAskPath
- CharReplace
- FormatMessage
- LogReadCustomNumber
- LogReadCustomString
- LogWriteCustomNumber
- LogWriteCustomString

You can copy this code from the InstallShield documentation, paste it into your InstallScript code, and customize it as necessary.

### **Expanded InstallScript Cabinet File Viewer**

The Cabinet File Viewer now provides additional information about .cab files for InstallScript projects.

- For features that were built by InstallShield, the viewer has the following new fields: Password Protected, Split Before, Split After, Split Not Allowed, and Split Before Not Allowed, and Image Index.
- For InstallScript Objects that are included in the InstallScript installation, the viewer has a new Object version field.
- For components, the viewer has the following new fields: Encrypted, Data as Files, and .NET Assembly.
- For media, the viewer has the following new field: Executable File.

## **Project Upgrade Alerts**

The following information describes possible upgrade issues that may occur when you upgrade projects that were created with InstallShield 2009 and earlier to InstallShield 2010. It also alerts you to possible changes in behavior that you may notice between new InstallShield 2010 projects and projects that are upgraded from InstallShield 2009 or earlier to InstallShield 2010. For updates to this information, see Knowledge Base article [Q200330](#).

## General Information about Upgrading Projects that Were Created in Earlier Versions of InstallShield

If you use InstallShield 2010 to open a project that was created with an earlier version, InstallShield 2010 displays a message box that asks you if you want to convert the project to the new version. If you reply that you do want to convert it, InstallShield creates a backup copy of the project with a file extension such as .768 before converting it. Delete the .768 part from the original project's file name if you want to reopen the project in the earlier version of InstallShield. Note that you cannot open InstallShield 2010 projects in earlier versions of InstallShield.

You can upgrade projects that were created with the following versions of InstallShield to InstallShield 2010: InstallShield 2009 and earlier, InstallShield 12 and earlier, InstallShield DevStudio, InstallShield Professional 7 and earlier, and InstallShield Developer 8 and earlier. Note that projects that were created with InstallShield MultiPlatform or InstallShield Universal cannot be upgraded to InstallShield 2010.

## Installing More than One Edition of InstallShield

Only one edition of InstallShield 2010—Premier, Professional, or Express—can be installed on a system at a time. Previously, it was possible to install the Express edition on the same system that had the Premier or Professional edition of the same InstallShield version.

## Change to the List of Supported Operating Systems for Running InstallShield

The minimum operating system requirement for systems that run InstallShield (the authoring environment) is now Windows XP or Windows Server 2003. Previously, the minimum operating system requirement was Windows 2000 SP3.

## Setup.exe No Longer Runs on Windows 9x, Windows NT 4, or Windows Me Systems

Setup.exe installations that are created in InstallShield can no longer be run on Windows 9x, Windows NT 4, or Windows Me. If an end user tries to launch Setup.exe on a Windows 9x or Windows Me system, Windows displays a message box with the following error: "The *FullSetup.exePathAndFileName* file expects a newer version of Windows. Upgrade your Windows version." On Windows NT 4 systems, Windows displays a message box with the following error: "*FullSetup.exePathAndFileName* is not a valid Windows NT application."

InstallShield no longer lists these legacy operating systems in any of the areas where target operating systems can be selected. For example, the Installation Requirements tab of the Project Assistant in Basic MSI and InstallScript MSI projects no longer lists these operating systems. In InstallScript projects, the Platforms tab on the Project Settings dialog box no longer lists these operating systems.

If you upgrade an InstallScript project that was created in InstallShield 2009 or earlier to InstallShield 2010, and if the operating system settings in the earlier project contained references to only these legacy operating systems, InstallShield replaces the legacy operating system options with the option for targeting all supported platforms.

## Windows Installer 1.x Redistributables Are No Longer Available

InstallShield no longer includes Windows Installer 1.x redistributables, since they target only legacy versions of Windows that are no longer supported. Previously, it was possible to add Windows Installer 1.x redistributables to a project through the Releases view, or through the Release Wizard.

## Redistributable for VBScript Runtime Files Is No Longer Available

InstallShield no longer includes the InstallShield object for VBScript Runtime Files. This redistributable targets only legacy versions of Windows that are no longer supported.

### InstallScript Dialog Source Code Changes

The InstallScript code for built-in InstallScript dialogs has been moved from individual InstallScript script files (.rul) to a single ISRTScriptDialogs.rul file. In addition, the event class drop-down list in the InstallScript view has a new Dialog Source option. If you select this option, the event handler drop-down list shows all of the built-in InstallScript dialogs. You can select any dialog in this list to customize its code.

InstallShield supports backwards compatibility: If you imported dialog source code into an InstallShield 2009 or earlier project and then you upgrade the project to InstallShield 2010, you can still use that dialog code. However, if you want to use the dialog sources that are now available when you select the dialogs in the event handler drop-down list, you must first make some changes to your upgraded project. Otherwise, you may encounter compile errors. In order to use the dialog code: On the Build menu, click Settings. On the Compile/Link tab, in the Preprocessor Defines box, enter the following:

```
_ISSCRIPT_NEW_STYLE_DLG_DEFS
```

After this definition has been added, if any dialogs had previously been imported into the project in earlier versions of InstallShield, the code in these imported .rul files may cause compile errors. These errors would need to be resolved.

If `_ISSCRIPT_NEW_STYLE_DLG_DEFS` is not defined, a warning message is displayed when the Dialog Source option is selected in the event class drop-down list in the InstallScript view.

The `_ISSCRIPT_NEW_STYLE_DLG_DEFS` definition is automatically added to all new projects that are created in InstallShield 2010.

This functionality applies to the following project types: InstallScript, InstallScript MSI, and InstallScript Object.

### InstallScript Header File Changes

The InstallScript header files (.h) have been reorganized. As a result, some of the .h files are obsolete. If an InstallScript file (.rul) in your project references one or more of these obsolete .h files, a compile warning is displayed whenever you compile your script or build a release. To resolve the warnings, remove any `#include` statements that reference the obsolete .h files, and rebuild the release. Also, ensure that `ifx.h` is referenced in an `#include` statement in your Setup.rul file or in other script files that are referenced by Setup.rul.

### InstallScript Changes for Windows API Prototypes

The service-related Windows API prototypes (such as `CreateServiceA`, `StartServiceA`, and `ControlService`) are now prototyped in `ISRTWindows.h` for all InstallScript, InstallScript MSI, and InstallScript Object projects. They are also now prototyped in `ISRTWindows.h` for all Basic MSI and Merge Module projects that include InstallScript custom actions.

If you want to use your own definitions for these Windows APIs instead of the ones that are now prototyped for InstallScript, add `ISINCLUDE_NO_SERVICEAPI` to the list of preprocessor definitions: On the Build menu, click Settings. On the Compile/Link tab, in the Preprocessor Defines box, enter `ISINCLUDE_NO_SERVICEAPI`. Otherwise, you may encounter compile errors.

Note that the constant `ISINCLUDE_NO_WINAPI_H` now suppresses only Windows API prototypes, not Windows constants or Windows structure definitions.

### Patch Creation (Basic MSI, InstallScript MSI)

InstallShield now uses the Windows Installer 4.5 patching technology to create patches. This change is reported for informational purposes.

## New Custom Actions and Database Tables for IIS Support in Basic MSI and InstallScript MSI Projects

If you add IIS support through the Internet Information Services view in a Basic MSI or InstallScript MSI project in InstallShield, InstallShield automatically adds several DLL custom actions to your project to support the IIS functionality. In InstallShield 2010, these custom actions have been enhanced to enable you to add applications to Web sites. In addition, these actions have been renamed to reflect standard naming conventions:

- ISIISSCosting—This custom action replaces the caExtractIISSuppFiles action.
- ISIISSRollback—This custom action replaces the caRlbackVRoots action.
- ISIISSUninstall—This custom action replaces the caRemoveVRoots action.
- ISIISSInstall—This custom action replaces the caCreateVRoots action.
- ISIISSCleanup—This custom action replaces the caIISCleanup action.

The entry points of each of the DLL custom actions have been renamed to match the corresponding custom action names.

Note that IIS functionality requires administrative privileges. Therefore, each of these DLL custom actions checks the Privileged property value. The value must be 1; if it is not, an error is displayed at run time. In InstallShield 2009 and earlier, each of the IIS custom actions had a Privileged = 1 condition. In InstallShield 2010, this condition is no longer set when you add new IIS Web sites or other IIS data to your project, since the custom actions check the Privileged property value.

If you upgrade a Basic MSI or InstallScript MSI project that contains IIS support from InstallShield 2009 or earlier to InstallShield 2010, InstallShield automatically updates and renames the custom actions accordingly. In addition, if the condition for an old custom action was not modified from the default condition, InstallShield also removes the Privileged = 1 condition. If a condition was modified, InstallShield leaves the existing condition as is. You can manually modify any of the conditions if appropriate.

In InstallShield 2010, all of the IIS data is stored in the ISIISSItem and ISIISSProperty tables. In InstallShield 2009 and earlier, the IIS data was stored in the following tables: ISIISSAppPool, ISIISSCommon, ISIISSMetaData, ISIISSWebServiceExtension, ISVRoot, ISVRootAppMaps, and ISWebSite. If you upgrade a Basic MSI or InstallScript MSI project that contains IIS support from InstallShield 2009 or earlier to InstallShield 2010, InstallShield automatically moves the IIS data to the new tables; InstallShield also deletes the old tables from the project.

## Changes for the Redistributables View

The Redistributables view has a new toolbar and group box area that provide robust search and organizational functionality. Use the new Show Details button in this view to show or hide the details pane for the selected redistributable in this view. The details pane provides information such as which files a redistributable installs. The Show Details button replaces the Show Details and Hide Details links that were previously available in the upper-right corner of this view.

The new group box area is below the new toolbar in the Redistributables view. You can drag and drop column headings onto this group box area to organize the list of redistributables in a hierarchical format. If you want InstallShield to separate all of the redistributables in the view into two groups—one whose check box is selected and one whose check box is cleared—drag the check box column to the group box area. This enables you to easily identify all of the redistributables that are included in your project. The result is similar to the behavior that previously occurred if you right-clicked any redistributable and then clicked Show Only Selected Items. Note that the Show Only Selected Items command is no longer available in the Redistributables view.

## Limiting the UI Level of a Chained .msi Package to that of the Main .msi Package

The UI level for a chained .msi package is now limited to be no higher than that of the parent package's current UI level. For example, in the following scenario, the chained .msi package is launched silently: you add a chained .msi package to your project in the Releases view and select Full UI (/qf) for its UI level setting, but the main installation is launched silently (/qn). Previously, the chained package showed exactly the UI level that was authored in the Releases view of the main installation; to restore this behavior, set the property ISChainExceedUILevel equal to the value 1.

## Encoding and Related Differences for XML File Changes

If you use the XML File Changes view to configure changes for a file that is already present on the target machine, or that is being installed as part of your installation, the installation now uses the encoding that is specified in that XML file, rather than the encoding that is specified in the XML File Changes view. This applies to new projects that are created in InstallShield 2010, as well as projects that are upgraded from InstallShield 2009 or earlier.

In addition, if the "Always create this element if it does not already exist" check box is cleared for an element that is not present in the target file, its child elements are no longer created. Thus, for an XML file such as //A/B/C, C is not created on the target system if B is neither present nor set to be created.

## Changes to the Major and Minor Version Registry Entries for the Uninstall Key of InstallScript Installations

InstallScript installations now create VersionMajor and VersionMinor registry values in the Uninstall key; the names of these values now match the entries that are created during Basic MSI and InstallScript MSI installations. This applies to new installations that are created in InstallShield 2010, as well as installations that are upgraded from InstallShield 2009 or earlier. Previously, in InstallShield 2009 and earlier, the names of the values that InstallScript installations created were MajorVersion and MinorVersion; these are no longer created.

In order to use the new registry values, the values of the following InstallScript constants have been changed:

- REGDB\_VALUENAME\_UNINSTALL\_MAJORVERSION is now VersionMajor instead of MajorVersion.
- REGDB\_VALUENAME\_UNINSTALL\_MINORVERSION is now VersionMinor instead of MinorVersion.

When the MaintenanceStart function is called, it creates the updated value names in the registry. By default, it also deletes the old value names if they exist. If you do not want the old value names to be deleted from target systems, you can use the new REGDB\_OPTIONS option called REGDB\_OPTION\_NO\_DELETE\_OLD\_MAJMIN\_VERSION.

If REGDB\_UNINSTALL\_MAJOR\_VERSION or REGDB\_UNINSTALL\_MINOR\_VERSION is used with the RegDBGetItem function, RegDBGetItem first checks for the new value; if the new value is found, the function returns the value data from the new value. If the new value is not found, the function automatically checks for the old value; if the old value is found, the function returns the value data from the old value.

To provide backwards compatibility, the following new constants are available:

- REGDB\_UNINSTALL\_MAJOR\_VERSION\_OLD
- REGDB\_UNINSTALL\_MINOR\_VERSION\_OLD

You can specify these constants with the RegDBGetItem, RegDBSetItem, and RegDBDeleteItem functions to get, set, and delete the old values.

The following new string constants are also available:

- REGDB\_VALUENAME\_UNINSTALL\_MAJORVERSION\_OLD is defined as MajorVersion.
- REGDB\_VALUENAME\_UNINSTALL\_MINORVERSION\_OLD is defined as MinorVersion.

## Removal of the SdShowMsg Dialog from the List of Editable Dialogs

The Dialog Editor does not currently support dialogs, such as SdShowMsg, that do not have a title bar; if you try to customize the SdShowMsg dialog, it may get corrupted. Therefore, this dialog is no longer displayed in the Dialogs view as one of the dialogs that you can edit. To customize this dialog, you should use the SdShowMsg call, not the Dialog Editor.

## Automation Interface Changes

If you use the automation interface with InstallShield or the Standalone Build, update your existing code to reflect the new ProgID: IswiAuto16.ISWiProject. The Standalone Automation Interface uses the same ISWiAutomation16.dll file that InstallShield uses, but it is installed to a different location.

Note that if you install the Standalone Build on the same machine as InstallShield, the last ISWiAutomation16.dll file that is registered is the one that is used.

The value of the eosAll constant for the OSFilter, which is a member of the ISWiComponent and ISWiRelease objects in the automation interface, has been changed. The new value is 64028880; previously, it was 5308624. If you are using the value of this constant to configure the list of operating systems for a component or a release through the automation interface, you must update your script to use the new value.

## Changes for the Locations of InstallScript Run-Time Script, Library, and Header Files

The InstallScript run-time library files that are installed with InstallShield have been consolidated into a central location, instead of several separate subdirectories. The script, library, and header files are now installed in Src, Lib, and Include folders in the following directory:

*InstallShield Program Files Folder\Script\Isrt*

The following folders are no longer installed, since the files within them are consolidated in the above location:

*InstallShield Program Files Folder\Script\IISRuntime*

*InstallShield Program Files Folder\Script\SQLRuntime*

*InstallShield Program Files Folder\Script\XMLRuntime*

Note that because of a name conflict, the Assert.h file in the following location is being renamed as ISAssert.h:

*InstallShield Program Files Folder\Script\Isrt\Include*

If you create a new project in InstallShield 2010, it uses the new locations of the files. If you upgrade a project from InstallShield 2009 or earlier to InstallShield 2010, InstallShield updates the projects to use the new locations.

## Changes in the Way that Linked Libraries and Their Locations Are Specified

The Compile/Link tab on the Settings dialog box has a new Additional Library Paths box that lets you specify the locations where the InstallScript compiler should search for InstallScript libraries (.obl files) that are not one of the standard InstallShield locations. For InstallScript and InstallScript Object projects, the standard locations are:

- <ISProductFolder>\Script\Ifx\Lib
- <ISProductFolder>\Script\Isrt\Lib

For Basic MSI and InstallScript MSI projects, the standard locations are:

- <ISProductFolder>\Script\Iswi\Lib
- <ISProductFolder>\Script\Isrt\Lib

If you create a new project in InstallShield 2010, InstallShield automatically lists the standard InstallShield script libraries such as Isrt.obl in the Libraries (.obl) box on the Compile/Link tab. However, InstallShield no longer includes the full path in that box. If you want to add your own custom libraries, you can specify the library file name in the Libraries (.obl) box, and the path in the Additional Library Paths box. You do not need to specify the full path and file name in the Libraries (.obl) box.

If you upgrade a project that was created in InstallShield 2009 or earlier to InstallShield 2010, InstallShield automatically removes the path of the standard script libraries that are listed in the Libraries (.obl) box.

### Removal of the InstallScript Structure Definition for ISOSVERSIONINFO

The definition of the ISOSVERSIONINFO structure, as well as the corresponding unused global instances of this structure, has been removed. The equivalent OSVERSIONINFO structure is still available. This definition removal does not cause any functionality changes; however, if you attempt to use the ISOSVERSIONINFO structure definition or global structure instances, a compiler error results.

To avoid a compiler error, do either of the following:

- Update the script to use the equivalent OSVERSIONINFO structure, and declare a local instance of this structure if needed. Update the script to use the appropriate structure member names. (Note that the OSVERSIONINFO member names are different than the ISOSVERSIONINFO member names.) Following is the definition of OSVERSIONINFO:

```
typedef OSVERSIONINFO
begin
    NUMBER nOSVersionInfoSize;
    NUMBER nMajorVersion;
    NUMBER nMinorVersion;
    NUMBER nBuildNumber;
    NUMBER nPlatformId;
    STRING szCSDVersion[128];
end;
```

- Declare the structure and structure instances locally as follows:

```
// Data structure that contains operating system version information.
// Used by ISCompareServicePack.
typedef ISOSVERSIONINFO // define a structure
begin
    LONG ISIOSVersionInfoSize; // Size in bytes of this data structure
    LONG ISIMajorVersion; // Major version number of the OS.
    LONG ISIMinorVersion; // Minor version number of the OS.
    LONG ISIBuildNumber; // Build number of the OS.
    LONG ISIPlatformId; // Operating system platform.
    STRING szISCSDVersion [128]; // Additional information about OS.
end;

// Variable for the operating system version information data structure.
// Used by ISCompareServicePack.
ISOSVERSIONINFO ISVersion;

// Pointer that points to the OS version information variable.
// Used by ISCompareServicePack.
ISOSVERSIONINFO POINTER pISVersion;
```

### Obsolete Keys Are No Longer Written to Setup.ini for InstallScript Projects

The following keys are obsolete and are no longer written to the Setup.ini file for InstallScript projects: Resource, EngineVersion, and EngineBinding.

### ISCab.exe Is No Longer Available

ISCab.exe is no longer supported. Therefore, it is no longer included with InstallShield.

### **Changes to Support for Securing Permissions for Files, Folders, and Registry Keys**

The General Information view has a new Locked-Down Permissions setting that lets you specify whether you want to use the new custom InstallShield handling or the traditional Windows Installer handling for all new permissions that you set for files, folders, and registry keys in your project. The new custom InstallShield handling option offers several advantages over the traditional Windows Installer handling option.

In all new projects, the default value for this setting is the custom InstallShield handling option. If you upgrade a project from InstallShield 2009 or earlier to InstallShield 2010, the traditional Windows Installer handling option is the default value of this setting.

This new setting is available in the following project types: Basic MSI, InstallScript MSI, Merge Module, MSI Database, MSM Database, and Transform.

### **Changes to the ReadyToInstall Dialog for Beta Windows Installer 5 Support of Per-User Installations**

The General Information view has a new Show Per-User Option setting. This setting lets you specify whether you want the ReadyToInstall dialog—in certain scenarios—to include buttons that let end users indicate how they want to install the product: for the current user or for all users. The per-user button sets the new Windows Installer property MSIINSTALLPERUSER equal to 1 to indicate that the package should be installed for the current user. The MSIINSTALLPERUSER property is available with the beta of Windows Installer 5.

If you create a new Basic MSI project in InstallShield 2010, the ReadyToInstall dialog includes support for the per-user and per-machine buttons; these buttons are displayed or hidden at run time if appropriate. If you upgrade a Basic MSI project from InstallShield 2009 or earlier to InstallShield 2010, the ReadyToInstall dialog does not have this support automatically. You can manually add these buttons and their associated conditions to the ReadyToInstall dialog if appropriate; use the ReadyToInstall dialog in a new InstallShield 2010 project as a guideline.

### **Public Directory Properties for Feature Destinations Are Added to SecureCustomProperties**

When you specify a location for the Destination setting of a feature and the location includes a public directory property, InstallShield now adds that property to the SecureCustomProperties property to allow end users to change the destination after the product has been advertised. This occurs in new projects that are created in InstallShield 2010. The change is also made for all feature destinations if you upgrade a project from InstallShield 2009 or earlier to InstallShield 2010.

This change applies to the following project types: Basic MSI and InstallScript MSI.

### **Changes to the Conditions for the InstallWelcome Dialog and the ResolveSource Action**

The condition on the InstallWelcome dialog and the ResolveSource action has been changed to "Not Installed" for all new Basic MSI projects that are created in InstallShield 2010. The conditions were changed so that the InstallWelcome dialog and the ResolveSource action can be used for a first-time installation with a patch. If you upgrade a Basic MSI project from InstallShield 2009 or earlier to InstallShield 2010, the conditions are not changed automatically. If you want the dialog and action to be used for a first-time installation with a patch, you can change the conditions in your upgraded project to "Not Installed".

### **Improvements to the .rtf File Size Limit for the SdLicenseRtf and SdLicense2Rtf Functions**

The file size limit for the .rtf files that are used with the InstallScript dialog functions SdLicenseRtf and SdLicense2Rtf is now 16 MB instead of 64 KB. Previously, if the file size was more than 64 KB, part of the EULA text was missing from the license dialog at run time.

Note that if you had overridden the SdLicenseRtf or SdLicense2Rtf functions in your script in InstallShield 2009 or earlier and then upgraded that project to InstallShield 2010, you would need to manually change the size limit by updating the SendMessage call with the EM\_EXLIMITTEXT message in DLG\_INIT. The iParam parameter (the fourth parameter) of the SendMessage call needs to be changed. The SendMessage call should be changed to this:

```
SendMessage( hEdi t, EM_EXLI MI TTEXT, 0, 0xffffffff );
```

Previously, the code contained this:

```
SendMessage( hEdi t, EM_EXLI MI TTEXT, 0, 0 );
```

### Removal of the OnResolveSource Event Handler from InstallScript MSI Installations

The InstallScript event handler OnResolveSource has been removed from InstallScript MSI projects. The Windows Installer now handles all source resolution. If you added the OnResolveSource event to an InstallScript MSI project in InstallShield 2009 or earlier and then you upgrade that project to InstallShield 2010, that event will no longer be called.

### Changes to the Way that a Log File Is Displayed from the SetupCompleteSuccess Dialog in Basic MSI Installations

The ShowMsiLog custom action now launches Notepad.exe from the SystemFolder directory, instead of from the WindowsFolder directory. Thus, if your installation is run on Windows Vista or later and the end user indicates on the SetupCompleteSuccess dialog that they want to view the log file, the installation launches Notepad.exe from the SystemFolder directory. This change was made because on Windows Server 2008 Standard Edition, Notepad.exe is available in the System32 directory, but not the Windows directory.

Note that behavior is available by default in all new Basic MSI projects that are created in InstallShield 2010. If you upgrade an InstallShield 2009 or earlier Basic MSI project to InstallShield 2010, InstallShield does not automatically change the behavior. You can manually change the behavior if necessary: In the Custom Actions and Sequences view, click the ShowMsiLog action. (If this action is not displayed, right-click the Custom Actions node and then click Show All Custom Actions.) Set the Filename & Commandline setting as follows:

```
[SystemFolder]notepad.exe "[MsiLogFileLocation]"
```

Thus, the value should contain [SystemFolder] instead of [WindowsFolder].

### Changes for ALLUSERS Property in InstallScript MSI Installations

Beginning with InstallShield 2010, the ALLUSERS property is set to 1 by default in all new InstallScript MSI projects. This is the recommended implementation, since most installations must be run in a per-machine context with administrative privileges. This value is also recommended to help avoid ALLUSERS-related issues when an InstallScript MSI installation is run silently.

If you upgrade a project that was created with InstallShield 2009 or earlier to InstallShield 2010, InstallShield does not automatically change the value of the ALLUSERS property or add this property if it was not defined in the earlier project.

### InstallScript Installations No Longer Include \_Setup.dll

InstallScript installations no longer include \_Setup.dll. Some earlier versions (DevStudio 9 and InstallShield X) did not log \_Setup.dll for uninstallation. As a result, this file was left behind in the Disk1 folder location (DISK1TARGET) after uninstallation. If an update was created with a later InstallShield version (InstallShield 10.5 through InstallShield 2009) and the update was for an original installation that was created with DevStudio 9 or InstallShield X, \_Setup.dll was deleted during uninstallation because \_Setup.dll was logged by the update. Since \_Setup.dll is not included in InstallScript installations that are created with InstallShield 2010, the \_Setup.dll file may now be left behind. Therefore, if you are updating from an installation that was created with DevStudio 9 or InstallShield X, you may need to delete the \_Setup.dll file manually (DISK1TARGET ^ "\_Setup.dll") during uninstallation to ensure that the uninstallation is complete.

## Saving a Project as an Earlier Version

InstallShield no longer has support for downgrading a project. That is, you cannot save an InstallShield 2010 project as an InstallShield 2009 or earlier project.

## Trialware Support

The only edition of InstallShield that includes the Trialware view is the Premier edition. This edition lets you create the Try and Die type of trialware. InstallShield no longer includes support for creating the Try and Buy/Product Activation type of trialware.

If you have an existing InstallShield Activation Service account and you want to be able to create the Try and Buy/Product Activation type of trialware in InstallShield 2010, you can still do so. For instructions, see Knowledge Base article [Q200884](#).

## Web Projects

The Web project type is no longer listed as one of the types of new projects that you can create in InstallShield. To use the same functionality that was available with a Web project in InstallShield 2009 and earlier, create a Basic MSI project, and then add a Web site in the Internet Information Services view.

The only difference between a Web project and a Basic MSI project was that a new Web project automatically contained a predefined folder for the IISROOTFOLDER directory in the Files and Folders view. All of the files that you add to the IISROOTFOLDER directory are installed to the Web server's root directory on the target system. InstallShield adds the predefined folder for the IISROOTFOLDER directory to a Basic MSI project when you add a Web site to the project. Thus, a Basic MSI project that contains at least one Web site configured in the Internet Information Services view is equivalent to a Web project that was created in InstallShield 2009 or earlier.

## Compact Projects

InstallShield no longer enables you to create new Compact projects. Note that if you created a Compact project in InstallShield 2009 or earlier, you can upgrade it to InstallShield 2010, and then make changes to it and build it. You can also use InstallShield 2010 to convert the Compact project to a Basic MSI project.

## Visual Studio Integration

Microsoft Visual Studio can be integrated with only one version of InstallShield Premier Edition or InstallShield Professional Edition at a time. The last version of InstallShield that is installed or repaired on a system is the one that is used for Visual Studio integration.

## Resolved Issues in InstallShield 2010 Original Release Version

### 1-12BN8C (Basic MSI, InstallScript MSI)

The InstallScript function SdShowMsg now displays the message box in front of Windows Installer dialogs when the function is called from an InstallScript custom action that is launched when the end user clicks the Next button. Previously, the message box was displayed behind Windows Installer dialogs.

### 1-14RXU1

If a string registry value that is configured in the Registry view contains square brackets ([ ]), the square brackets are now created correctly in the registry at run time. Previously, a registry value of [1], for example, would be set as [\[]1[\]] at run time.

### 1-157P71 (InstallScript MSI)

The cached .msi database is no longer left on the target system during uninstallation.

## 1-1A1R0H

The InstallShield Help Library now has a new help topic called, "Using the Automation Interface on a 64-Bit System"; this help topic explains that if you are using the automation interface on a 64-bit machine, you may need to load the automation interface through a 32-bit executable file. For example, if you are using VBScript with the automation interface, you may need to launch cscript.exe from the 32-bit system folder (SysWow64). Otherwise, the 64-bit scripting host may encounter an error when creating the automation object.

## 1-1AEXPD

The Component Wizard help now explains how to access the Component Wizard.

## 1-GK9AV (Basic MSI)

When an InstallScript dialog is displayed through an InstallScript custom action, the Cancel button now works. Previously, clicking the Cancel button at run time had no effect.

## 1-J9GQL (InstallScript)

An installation no longer fails to end if SdShowMsg is being displayed when abort; is called.

## 1-NGXU5

The InstallScript compiler now enables you to use a maximum of about 4,294,967,295 statements instead of only 65,535 statements in setup.inx, the compiled script file. If this limit is exceeded, error -5009 may occur during setup initialization.

Following are additional limits for a compiled script:

- Total number of global variables: about 196,605 (65,535 numbers, 65,535 variants, 65,535 strings)
- Total number of typedefs: about 65,535
- Total number of prototypes: about 65,535
- Total number of functions: about 65,535
- Total number of statements per function: about 65,535
- Total local variables per function: about 196,605 (65,535 numbers, 65,535 variants, 65,535 strings)

Following are limits for script files (.rul):

- Maximum line width: 1,024 characters
- Maximum number of nested include files: 80
- Total number of include files: 2,048
- Maximum identifier length: 63 characters
- Maximum number of macro expansions: 100
- Maximum macro expansion text length: 256 characters
- Maximum file name length: 256 characters
- Maximum number of nested #if statements: 10
- Maximum parameters per function: 16

Compile errors occur if one or more of these .rul limits are exceeded.

## 1-OIH93

All of the setting names are translated in the Shortcuts view of the Japanese version of InstallShield. Previously, one of the setting names was displayed in English.

## 1-WNQYL

The German translation for the Login ID control on the SQLLogin run-time dialog is no longer truncated.

## 1-YTJSK

The LoadStringFromStringTable help topic was corrected.

## IOA-000027227, IOB-000004430

If you use the XML File Changes view to configure changes for an XML file that has a DOCTYPE declaration, run-time error 27520 no longer occurs. In addition, the DOCTYPE/DTD information is no longer removed from the XML files at run time.

## IOA-000027674, IOC-000051784, IOC-000074857 (Basic MSI, InstallScript MSI, QuickPatch)

A Val0004 validation error now occurs only in the following cases:

- The key file of a component in the previous package has a higher version than the key file of the new package.
- The key file of a component in the previous package has a version, but the same file in the new package does not.

This validator verifies that when the upgrade is applied, all files that have changed in the upgrade will be updated properly on the target system. Previously, Val0004 occurred erroneously in several scenarios when validation was performed for an upgrade:

- You added a new file to a component but you did not change the key file.
- You changed a file in a component, but you did not change the key file for that component.
- A file in a component was marked as a companion file, and it was different from the file in the previous package. However, you did not change the companion parent file.

In all of these previous scenarios, Windows Installer updated the new or modified file during an upgrade, even though the Val0004 indicated that it would not.

Also as part of this change, Val0010, which was similar to Val0004, has been removed. Previously, Val0010 occurred erroneously in several scenarios when validation was performed for an upgrade.

The Upgrade Validation Wizard no longer encounters an unhandled exception when Val0004 is run.

For detailed information on how Windows Installer determines whether a file should be overwritten, see "Overwriting Files and Components on the Target System" in the InstallShield Help Library. This help topic has been corrected.

## IOA-000028856 (Merge Module)

If you enter a value in the Template Summary setting for a product configuration in the Releases view of a Merge Module project, that value now overrides the value that was entered in the General Information view. Previously, the value in the General Information view was always used.

## IOA-000029522, IOC-000072345 (Basic MSI, InstallScript MSI)

Build warning -6248 no longer occurs for a file called .dll or one of its dependencies.

## IOA-000031823 (InstallShield MSI Log Analyzer)

The InstallShield MSI Log Analyzer no longer generates a Type Mismatch error when the Run, Log & Analyze button is clicked for a detailed log report. Previously, this error occurred for some log files.

### **IOA-000032253 (Basic MSI, InstallScript MSI)**

The Device Driver Package panel in the Device Driver Wizard has a new System Architecture list. If the device driver package (.inf) includes support for multiple system architectures or a specific one, use the new System Architecture list to specify the one that you want to target. Previously, if an .inf file contained multiple SourceDisksFiles sections (for example, one for AMD64, one for x86, and one that is platform independent), only the files in the platform-independent section were scanned.

### **IOA-000032365 (InstallScript MSI)**

If you create an uninstallation shortcut through the Application Shortcuts page of the Project Assistant in an InstallScript MSI project, the uninstallation shortcut now removes the product's Add or Remove Programs entry on the target system. Previously, the entry was not removed.

### **IOA-000036623**

SYSINFO.WINNT.bWinXP no longer returns true for Windows Server 2003 Small Business Edition.

### **IOA-000039055 (Basic MSI, InstallScript MSI)**

The Windows Installer version that is selected for a release on the Setup.exe tab in the Releases view is now included in the release if you build it through the Build Installation page in the Project Assistant. Previously, building a release through the Project Assistant always included version 2 of the Windows Installer, regardless of what version was selected in the Releases view.

### **IOA-000040847 (InstallScript, InstallScript MSI)**

The file size limit for the .rtf files that are used with the InstallScript dialog functions SdLicenseRtf and SdLicense2Rtf is now 16 MB instead of 64 KB. Previously, if the file size was more than 64 KB, part of the EULA text was missing from the license dialog at run time.

Note that if you have overridden the SdLicenseRtf or SdLicense2Rtf functions in your script in InstallShield 2009 or earlier and then upgraded that project to InstallShield 2010, you would need to manually change the size limit by updating the SendMessage call with the EM\_EXLIMITTEXT message in DLG\_INIT. The iParam parameter (the fourth parameter) of the SendMessage call needs to be changed. The SendMessage call should be changed to this:

```
SendMessage( hEdi t, EM_EXLIMITTEXT, 0, 0xffffffff );
```

Previously, the code contained this:

```
SendMessage( hEdi t, EM_EXLIMITTEXT, 0, 0 );
```

### **IOA-000040961**

If you install InstallShield on a machine that already has the olelib.tlb file in the System32 folder, the InstallShield installation no longer overwrites that version.

### **IOA-000041856 (Basic MSI, InstallScript MSI)**

If you add to a project an InstallScript custom action that is scheduled in the Install Execute sequence after InstallInitialize, the custom action no longer unexpectedly closes the MSI handle when the installation is run. Previously in some cases, the log file showed that the action unexpectedly closed the handle.

### **IOA-000042190 (InstallScript Object for NT Services)**

The text fields in the NT Services Wizard are no longer black.

### **IOA-000042725 (InstallScript MSI)**

If you build a patch for a multilingual InstallScript MSI installation, the patch user interface is now displayed in the language that was used for the interface of the original base installation. Previously, the patch user interface was displayed in English, regardless of what language was used for the original base installation.

### **IOA-000042919, IOC-000044383**

The GetMemFree documentation now states that the InstallScript function GetMemFree is obsolete. It also says that GetSystemInfo should be used to determine the amount of physical memory that is available on the target system.

### **IOA-000043029 (Merge Module)**

InstallShield no longer crashes when you use the Advanced Settings area for a component to add a device driver.

### **IOA-000043054 (Basic MSI, InstallScript MSI)**

Build error -3204 no longer occurs if you have not specified an icon index for a shortcut.

### **IOA-000043582**

If you use the General Information view to remove a setup language from a project and then add that language back, all custom string entries are now included for that language. Previously in that scenario, the custom string entries were not added for the language unless a language that was never included in the project was added.

### **IOA-000044469 (Basic MSI, InstallScript, InstallScript MSI)**

When you import an xml element that contains an unused namespace declaration, that declaration is now imported to the Namespace tab for an XML file in the XML File Changes view. Previously, since this mapping was not used in any of the XML elements being imported, it was ignored; if you wanted to use the mapping, it was necessary to manually add the namespace in the XML File Changes view.

Note that if the prefix is not used for any elements or attributes that are created or modified on the target system, the installation does not add the namespace declaration to the XML file at run time.

### **IOA-000043838 (InstallScript, InstallScript MSI, InstallScript Object)**

When you are using the Dialog Sampler to preview run-time dialogs, the vertical position of the dialogs remains consistent. Previously, some dialogs were displayed at slightly different vertical positions.

### **IOA-000043934**

The "Using Bit Flags" help topic no longer contains incorrect samples. Previously, the portions that explained how to clear a bit flag from a variable were incorrect.

### **IOA-000044115 (Basic MSI, InstallScript MSI)**

When UAC is disabled on a Windows Vista system, the end user is using a limited user account, and the installation needs to install an InstallShield prerequisite that requires elevation, the installation now displays a message box to inform that elevated privileges are required. The message box includes Retry and Cancel buttons.

Previously, the installation required the end user to click the Install button or the Cancel button. In this case, if the end user clicked the Install button, the same dialog was displayed again, prompting the end user to click the Install button or the Cancel button.

### **IOA-000044257, IOC-000074136 (InstallScript)**

The Export Components Wizard command is no longer available when you right-click the Setup Design node in the Setup Design view of InstallScript projects, which do not include support for the wizard.

### **IOA-000044505 (QuickPatch)**

The SetupStatus dialog displays correct version details when a QuickPatch package is applied to a target system and the QuickPatch is for an InstallScript MSI project. Previously, this dialog displayed the wrong version number.

### **IOA-000044607 (Basic MSI, InstallScript MSI, Merge Module)**

The predefined system search for the .NET Framework 3.0 SP1 has been corrected; it is now searching in the appropriate registry location on the target system.

### **IOA-000044659, IOA-000044761, IOC-000074756 (QuickPatch)**

InstallShield no longer crashes when you build a QuickPatch package that updates .NET assembly files.

### **IOA-000044799 (InstallScript)**

InstallShield no longer crashes when you test some XML file changes from within the XML File Changes view and then build a release.

### **IOA-000045502 (Merge Module)**

ISICE04 no longer occurs when you use the Certified for Windows Vista Merge Module Validation Suite for a merge module that contains ODBC data.

### **IOA-000045583 (Merge Module)**

The maximum number of characters in the name of a custom action in a merge module project is now 27, instead of 35. This limit enables the custom action to be sequenced in the merge module project.

### **IOA-000045856 (Basic MSI, InstallScript MSI)**

If the Template Summary and product language are set to a non-English value and the installation is run on an x64 target machine, an InstallScript custom action in that installation no longer fails with an unknown exception.

### **IOA-000045889 (Basic MSI, InstallScript, InstallScript MSI)**

If you use the ExportStrings method of the ISWiProject object of the automation interface and you specify a valid date for the dTimeStamp parameter, the automation interface now exports only the strings that have been modified since the date that you specified. Previously, the automation interface exported all of the strings.

### **IOA-000045891 (Basic MSI, InstallScript MSI)**

When the trademark symbol is used with the product name, it is now displayed correctly on the setup prerequisite dialog. Previously, it was displayed as a block character.

### **IOA-000045950 (InstallScript MSI)**

Registry keys that are installed during the OnInstallFilesActionAfter event are now removed during uninstallation.

### **IOA-000046415 (Basic MSI)**

Feature prerequisites can now be launched from paths that contain Unicode strings.

### **IOA-000046527 (Basic MSI, InstallScript MSI, Merge Module)**

It now takes less time for COM+ applications that were configured in the Component Services view to be installed on target systems. Previously, a performance issue existed with COM+ application installations that were built in InstallShield 2008 and 2009.

### **IOA-000047262 (Basic MSI, InstallScript MSI)**

If you right-click a merge module custom action in one of the sequences in the Custom Action and Sequences view and then click Remove, InstallShield now removes that action from the sequence. Previously, the action was not removed from the sequence.

### **IOA-000047571 (Basic MSI, InstallScript MSI)**

One of the run-time message boxes that may be displayed in certain scenarios for an InstallShield prerequisite now contains an OK button, instead of Yes and No buttons. The message box explains that the files for the InstallShield prerequisite could not be found—probably because of a failed or canceled download—and that the installation will stop.

#### **IOA-000047619 (Basic MSI, InstallScript MSI)**

If an installation that contains an InstallShield prerequisite is run from a network location, it now displays the "Open File - Security Warning" dialog only once. Previously, the installation displayed it two or more times: once as the Setup.exe file was launched, again after the end user clicked the Install button, and then possibly again for each prerequisite that was launched.

#### **IOA-000047743 (Basic MSI, InstallScript MSI)**

If either of the .NET Framework 3.5 SP1 prerequisites needs to be downloaded at run time, they can now be downloaded and installed properly, and progress is shown. Previously, since progress was not shown, it may have seemed that the download did not finish. In addition, the download for the Web download prerequisite may have failed.

#### **IOA-000047792 (Basic MSI, InstallScript MSI)**

You can now enter a fourth field in the Product Version setting of the General Information. In InstallShield 2009, only three fields could be entered.

#### **IOA-000047983 (Basic MSI)**

If you edit radio buttons on a dialog in a multilanguage project and the dialog is not the default project language, the radio buttons are no longer rearranged to be overlapping each other.

#### **IOA-000048062 (Standalone Build)**

The installation for the Standalone Build now installs LogUtil.dll. Previously, this file was not installed for the Standalone Build; therefore, build error -1024 may have occurred if the Standalone Build was used to build a release in some scenarios.

#### **IOA-000048079 (Basic MSI, InstallScript MSI)**

The Russian version of the setup prerequisite dialog now includes single quotes instead of double quotes where appropriate.

#### **IOA-000048143 (Basic MSI, InstallScript MSI)**

The text on the Japanese version of the setup prerequisite dialog is no longer truncated. Previously, if the product name was long, the text would wrap to a third line, and part of the text on the third line was cut off.

#### **IOA-000048544 (InstallScript MSI)**

If you select the New Style option for the InstallScript User Interface Type setting in the General Information view, the installation creates the directory structure as it is configured in the Files and Folders view. Previously in some cases, some issues occurred. For example, some directories were created as child directories instead of as sibling directories.

#### **IOA-000048693 (Basic MSI, InstallScript, InstallScript MSI)**

The behavior of the buttons on the XML Element panel in the Import XML Settings Wizard have been improved. These two buttons—Select Branch and Clear Branch—now enable you to select or clear the check boxes of only the selected element and its children. Previously, the buttons selected or cleared the check boxes for all of the elements in the entire XML file, regardless of what element was selected in the file. (Note: To achieve the same

results as the old behavior, select the root node of the XML file, and then click the Select Branch button or the Clear Branch button.)

#### **IOB-00004206**

The SdLogonUserBrowse help topic now states that the SdLogonUserBrowse dialog is displayed when an end user clicks the Browse button on the SdLogonUserInformation dialog.

#### **IOA-000048241 (InstallScript, InstallScript MSI)**

The letter *P* is no longer used as a keyboard shortcut twice in the default French version of the SdSetupType2 dialog.

#### **IOA-000048627**

The correct Swedish translation for the Cancel button is now used on Swedish run-time dialogs.

#### **IOA-000048734 (InstallScript MSI)**

If No is selected for the Enable Maintenance setting in the General Information view, the installation no longer aborts and displays an error. Previously, the installation ended and displayed an error message stating that the setup had not detected an installed version of the product, and that the specified command-line option required that the product be installed.

#### **IOB-000050682 (InstallScript, InstallScript MSI)**

When an InstallScript or InstallScript MSI installation sets the value of the UNINSTALL\_STRING variable, it now includes a space before the uninstall string text substitution value.

#### **IOB-000051080**

If FILE\_LOCKED is used with the InstallScript function Is and the file on the target system is not accessible because of insufficient privileges, the function now returns TRUE.

#### **IOB-000052648 (InstallScript)**

If you specify the same preprocessor definition on the Build tab in the Releases view and on the Compile/Link tab of the Settings dialog box (which is displayed when you click Settings on the Build menu), a compile error no longer occurs.

#### **IOB-000053433 (InstallScript)**

If you build a release that contains a large number of dynamically linked files and some of the files have long file names, build error 110 no longer occurs. Previously, this build error occurred under certain conditions.

#### **IOB-000054168**

The trial extension confirmation dialog that is displayed for trialware now shows the number of days that are left in the trial period on Danish systems. Previously, this dialog displayed a format specifier instead of the number of days.

#### **IOB-000055577 (InstallScript MSI)**

The InstallShield Premier edition now lets you use the New Language Wizard to add unsupported languages to InstallScript MSI projects.

#### **IOB-000055736, IOC-000077622 (InstallScript MSI)**

When the setup launcher installs a new Windows Installer engine at run time, the rest of the installation uses that new version of Windows Installer. Previously, the earlier version that existed on the target system was used for the rest of the installation; in some cases, this caused the installation to fail.

#### **IOB-000055960**

InstallShield displays Russian characters correctly for Russian dialogs in the Dialogs view. Previously, if you selected a control that had text, the Text field for that control showed question marks instead of Russian characters.

#### **IOB-000055980 (Basic MSI, InstallScript MSI)**

The Windows Mobile Wizard and the Smart Device Setup Wizard now contain options for specifying target versions of Windows Mobile 5.x/6.x with different processors for .NET Compact Framework and SQL redistributables. Previously, the options did not list the 6.x version. Note that the .cab files support both 5.x and 6.x platforms.

#### **IOB-000056122**

Visual Studio no longer crashes when dynamic scanning is being performed from within an InstallShield project that is opened from within Visual Studio. Previously, this may have occurred if Data Execution Prevention (DEP) was turned on.

#### **IOB-000056456 (InstallScript MSI)**

When a compressed release is built for a multilanguage project that includes a different splash screen for each language, the appropriate language splash screen is now displayed at run time. Previously, the wrong language splash screen was displayed.

#### **IOB-000056808 (Basic MSI)**

InstallShield now lets you add to a project files that are in a directory whose name has nine or more two-byte characters. Previously, InstallShield crashed in this scenario.

#### **IOB-000056969**

The Danish translation of a string on the SetupCompleteSuccess dialog has been corrected.

#### **IOB-000057048 (Basic MSI, InstallScript MSI)**

You can now use the System Search view to define a search for data in an XML file that has a DOCTYPE declaration. Previously, the system search was not successful at run time if the XML file had a DOCTYPE declaration.

#### **IOB-000057338 (InstallScript, InstallScript MSI)**

The InstallScript system variable SHELL\_OBJECT\_FOLDER is now initialized to the same value as IFX\_PRODUCT\_NAME only if the installation is not running in maintenance mode. Any changes that are made to SHELL\_OBJECT\_FOLDER later in the script are logged and are read from the log on subsequent maintenance operations.

#### **IOC-000005958 (Basic MSI, InstallScript MSI)**

If you use braces ({} ) and a Windows Installer property in a field in the XML File Changes view, the data is now correctly formatted at run time per Windows Installer formatting rules. Previously, leading braces resulted in invalid formatting results.

#### **IOC-000041388, IOC-000062140 (Basic MSI, InstallScript MSI)**

If you select a feature in the "Add new components to the feature" list in the Files and Folders view and then configure permissions for a folder, InstallShield creates a component for folder permissions and associates it with the selected feature. Previously, InstallShield did not associate the component with the selected feature.

## **IOC-000043917, IOC-000058435, IOC-000059687, IOC-000063555, IOC-000071845 (Transform)**

If you use the Direct Editor in a transform project to add a new row to a table and then change the primary key of that row, the Direct Editor now marks the row as added. Previously, the Direct Editor marked the row as deleted and added a new additional row.

In addition, if you use the Direct Editor in the transform project to delete a row from a table, the Direct Editor now marks the row as deleted. Previously, the Direct Editor removed the row from the table.

The Direct Editor also enables you to re-add a table row to an .mst file. Previously, once a row was removed from the transform, it was not possible to re-add it.

## **IOC-000046889 (InstallShield MSI Log Analyzer)**

The InstallShield MSI Log Analyzer no longer crashes under certain conditions.

## **IOC-000046945**

If you press F1 in the InstallScript view when the cursor is within the StrReplace function, the "StrReplace" help topic is displayed. In addition, this function is listed in the "String Functions" help topic.

## **IOC-000051726 (Basic MSI, InstallScript MSI)**

If the .NET Framework is added to a project through the Releases view and it is configured to run without the user interface, the progress bar is now updated while the .NET Framework is being installed. Previously, the progress bar was not updated.

## **IOC-000052935, IOC-000073753, IOC-000076417 (InstallScript MSI)**

If you use InstallShield 2010 to create an InstallScript MSI installation and a major upgrade for that installation, the major upgrade now removes the Add or Remove Programs entry for the base installation. It also removes any InstallScript binaries of the base installation from the [ProgramFilesFolder]InstallShield Installation Information directory.

## **IOC-000053525 (Basic MSI, InstallScript MSI)**

If you use a property name and some other text to set an attribute value in the XML File Changes view, and if the property is undefined at run time, the installation sets the attribute value equal to the other text. Previously in that scenario, the installation did not set the attribute value.

## **IOC-000053961 (Basic MSI)**

A single quote mark in a string for a control event condition is now displayed correctly. Previously, if a single quote mark was used, it was not displayed correctly, and sometimes the Custom Actions and Sequences view could not be displayed.

## **IOC-000054341 (Basic MSI, InstallScript, InstallScript MSI)**

In the XML File Changes view, you can add an attribute to an element more than once, and configure installation changes for one instance of that attribute, and uninstallation changes for the other instance. At run time, the installation and uninstallation changes are made for the attribute as expected. Previously, only one arbitrary change per attribute name was applied at run time, and only if its scheduling matched the installation operation.

## **IOC-000054448 (InstallScript MSI)**

If you choose to modify the InstallScript event handler OnAbort in the InstallScript view, this event handler is now added to your script with the correct default code. Previously, no code was added.

## **IOC-000054833**

The DoInstall and LaunchApplication help topics now mention that you may need to use the /clone\_wait parameter for launching child installations from removable media. Otherwise, the installation may not wait for the child installations to complete.

#### **IOC-000055098 (Basic MSI, InstallScript MSI)**

If you use the BuildPatchConfiguration method of the automation interface to build a major-upgrade patch, the patchwiz.dll confirmation dialog box that asks whether you want to proceed is now suppressed.

#### **IOC-000055311 (InstallScript, InstallScript MSI)**

If you call ChangeDirectory(WINDIR); in an OnEnd event, an Explorer window is no longer launched when the product is uninstalled from Add or Remove Programs. Previously, this occurred in some cases on 64-bit systems.

#### **IOC-000056864 (InstallScript)**

If the path to the installation contains unsupported characters, the installation now displays an appropriate error message instead of crashing.

#### **IOC-000057374 (InstallScript, InstallScript MSI)**

If you override the TextStyle setting to change the font or font size of the text on a run-time dialog, the text is now displayed with the new text style. Previously, for the Japanese version of InstallShield, the default text style was used instead of the new text style.

#### **IOC-000057999 (InstallScript MSI)**

SdFeatureTree now properly sets feature states based on a recorded response file.

#### **IOC-000058568 (Standalone Build)**

ReleasePackager.exe is now installed with the Standalone Build.

#### **IOC-000058825, IOC-000072576 (Basic MSI, InstallScript MSI)**

If an InstallScript batch file function in an InstallScript custom action is scheduled for deferred, commit, or rollback execution, InstallScript variables are not set to Windows Installer properties. Previously, the variables were set to Windows Installer properties. However, deferred, commit, and rollback custom actions do not have access to Windows Installer properties; therefore, the custom actions failed.

#### **IOC-000059065, IOC-000067722 (Basic MSI, InstallScript MSI)**

The icons for custom actions that are under the Custom Actions node in the Custom Actions and Sequences view are now visually different, depending on the scheduling of the action. Icons for deferred actions have a blue dot, icons for commit actions have a green dot, and icons for rollback actions have a red dot. Icons for immediate actions do not include a dot.

#### **IOC-000059123, IOC-000074332 (InstallScript, InstallScript MSI)**

If billboards are included in a project, they are now displayed in front of the background window at run time. Previously, they were displayed behind the background window.

#### **IOC-000059177 (MSI Database)**

The value of the ScrollableText control can now be changed in direct edit mode. Previously, changing the value did not update the Windows Installer tables.

#### **IOC-000060997 (Basic MSI, InstallScript MSI)**

If you are using the self-hosted solution for FLEXnet Connect and you configure the Host setting in the Update Notifications view, the Is Product/Version Registered setting in this view now works correctly. That is, if the product and version are not registered, this setting displays No. If they are registered, this setting displays Yes. If you click the Refresh button for this setting, InstallShield checks the host to see if they are registered and updates the Is Product/Version Registered setting as appropriate. Previously, the setting always displayed No for the self-hosted solution, even if the product and version were registered.

### **IOC-000061105 (Basic MSI, InstallScript MSI)**

The options that are available for the Wait Type setting of a control NT service component have been reversed. The default option is now "Wait for the event to complete"; this option corresponds with a value of 1 for the Wait column of the ServiceControl table. The "Wait for the SCM" option now corresponds with a value of 0 for the Wait column of the ServiceControl table.

### **IOC-000062077**

The Select String dialog box is displayed when you click the ellipsis button in any Text property for a dialog control in the Dialogs view. This dialog box now lets you sort the strings that are displayed on this dialog box by any column.

### **IOC-000062387 (Basic MSI, InstallScript MSI)**

SQL logins no longer fail when connecting to a remote database server with Windows authentication.

### **IOC-000063280 (Basic MSI, InstallScript MSI)**

InstallShield no longer removes the backslash after a directory property in a path that is specified in the Target setting for a shortcut. Previously, the backslash was automatically removed, which was not appropriate for properties such as [SETUPEXEDIR] and [SUPPORTDIR].

### **IOC-000063475 (InstallScript MSI)**

InstallScript MSI installations that target 64-bit target systems now create the uninstall key in the 32-bit section of the registry, even if you include the REGDB\_OPTION\_WOW64\_64KEY option in REGDB\_OPTIONS. Previously, if the REGDB\_OPTION\_WOW64\_64KEY option was used, the installation created the uninstall key in the 64-bit section of the registry, which caused an error during uninstallation.

### **IOC-000063747**

If you add a new string identifier and value to one of the languages in a multilanguage project, InstallShield adds that string identifier and value to all of the other languages in your project. You can override the string value for each language if appropriate. Previously, the string identifier was added to all of the other languages, but the value was left blank. In some cases, if you did not enter a string value for all of the other languages, a build error occurred.

### **IOC-000064551, IOC-000055096 (Basic MSI)**

The condition on the InstallWelcome dialog and the ResolveSource action has been changed to "Not Installed" so that they can be used for a first-time installation with a patch. Previously, the condition for the dialog was "Not Installed And (Not PATCH Or IS\_MAJOR\_UPGRADE)"; the condition for the action was "Not Installed And Not PATCH".

### **IOC-000065010, IOC-000078120 (InstallScript MSI)**

If an InstallScript MSI installation is configured to be 64 bit (that is, the template summary property is set to a 64-bit value), a full set of registry values is now written to the 64-bit part of the registry. Previously, some of the Add or Remove Programs information was not written to the 64-bit part of the registry.

### **IOC-000067072 (InstallScript)**

Difxapi.dll, which is used for installing device drivers when DIFx support is enabled, is now deleted from the Temp folder on the target system when the installation is finished.

### **IOC-000067779 (Basic MSI)**

Clicking the Cancel button when IIS data is being installed now successfully rolls back the installation. Previously in that scenario, clicking the Cancel button did not have any effect.

### **IOC-000067781 (InstallScript)**

A splash screen is now displayed for a minimum of 5 seconds if no value is entered for the Minimum Initialization Time setting for a release in the Releases view or for the SplashTime property in the Setup.ini file. Previously, the splash screen was displayed so briefly that it sometimes seemed like the splash screen was not displayed at all.

### **IOC-000068993 (InstallScript)**

You can use the new InstallScript function SetObjectPermissions to set permissions for the new user that is created through the SdLogonUserInformation function.

### **IOC-000069095 (Basic MSI)**

If the Windows Installer 3.1 is added to a release in the Releases view, and if an InstallShield prerequisite in the project is configured so that Windows Installer must be installed before the prerequisite, the target system is now restarted after Windows Installer is installed. Previously, the target system was not restarted.

### **IOC-000069216, IOC-000071202 (Basic MSI, InstallScript MSI)**

If a language is selected in the UI Languages setting for a release in the Releases view and then you remove that language from the Setup Languages setting in the General Information view, you can now remove that language from the release. In the UI Languages setting, the language is displayed with a red exclamation mark; you can clear its check box to remove the language. Previously, it was necessary to first remove the language from the UI Languages setting and then remove it from the Setup Languages setting; otherwise, the UI language would still be built into the release.

If you build a release in which a particular language is selected in the UI Languages setting but not in the Setup Languages setting, build error -7201 now occurs to inform you that the language is missing from the project.

### **IOC-000069256 (Basic MSI)**

If your project includes multiple-instance support, you can now override the INSTALLDIR property. Previously, overriding this property caused run-time error 1606.

### **IOC-000070648, IOC-000070652 (Basic MSI, InstallScript MSI)**

When you use the Internet Information Services view to add a Web site, virtual directory, and Web service extension to your project, and you build a release and perform validation for it, ICE32 errors ("possible mis-aligned foreign keys") no longer occur.

If your project has IIS tables that contain one or more references that are not present in the Component table, an ICE03 error ("not a valid foreign key") occurs during validation. Previously, this error did not occur in this scenario.

### **IOC-000070072**

If you activated InstallShield, the About InstallShield dialog box, which is displayed when you click About InstallShield on the Help menu, now has a Return License button. You can click this button to return your license to your account on the activation server. Note that if you do this without uninstalling InstallShield, InstallShield

reverts back to trial mode if any days remain in your trial period. If the trial period has ended, InstallShield will stop working.

#### **IOC-000070176 (Basic MSI, InstallScript MSI)**

InstallShield no longer stops responding when you specify a previous version of your product in the Patch Design view. Previously in this case, InstallShield stopped responding under certain conditions.

#### **IOC-000070649 (Basic MSI, InstallScript MSI)**

Release flags now filter IIS data. Previously, if you assigned a release flag to a feature that contained IIS data and then built a release that was configured to exclude that release flag, InstallShield still included the IIS data in the built release.

#### **IOC-000070931**

The sample code in the documentation for the DefineDialog and EzDefineDialog dialog functions has been corrected. It now contains an if-then statement that allows the dialog's Close and Help buttons to work.

#### **IOC-000070941 (InstallScript)**

Build warning -7203 now occurs if the product version exceeds the maximum version that is allowed (255.255.65535). Previously, no build warning occurred. Although it may be possible to install an InstallScript installation whose product version exceeds the limit, maintenance and other operations may result in a run-time error.

#### **IOC-000071245 (Basic MSI, InstallScript MSI)**

When a project is upgraded to a new version of InstallShield, InstallShield adds only one instance of each property in the SecureCustomProperties property. Previously in this scenario, InstallShield added properties to SecureCustomProperties when the project was upgraded to the new version of InstallShield, even if the properties were already included.

#### **IOC-000071503 (Basic MSI, InstallScript, InstallScript MSI)**

Custom HTTP errors that are configured in the Internet Information Services view can now exceed 255 characters. Previously, if any of the errors exceeded 255 characters, that error message was truncated when it was displayed on the target system, and the remaining errors in the installation were not shown.

#### **IOC-000071652**

SYSINFO.WINNT.bWinServer2003 now returns true for a Windows Server 2003 system that is a domain controller.

#### **IOC-000071657 (InstallScript, InstallScript MSI)**

If you are running an installation in record mode to create a response file, the return value of SelectDir and SelectDirEx is now captured correctly. Previously, the Setup.iss file recorded the number 0, even if the OK (1) or Cancel (2) button was clicked.

#### **IOC-000071729 (InstallScript)**

If No is selected for the Uninstall setting of components that contain IIS data (such as application pools, Web sites, and virtual directories), the IIS data are now left permanently on the target system; they are no longer uninstalled when the product is uninstalled.

#### **IOC-000071732**

The Open MSI/MSM Wizard can successfully convert an .msi database to an InstallShield project file (.ism), regardless of the column width of the Sequence column in the File table. Previously, if the column width was not i2, the wizard created a new project file with no records in it.

#### **IOC-000071908**

Visual Studio 2008 solutions that contain a database project now display the data in InstallShield views correctly. Previously, views—such as the Files and Folders view—that contained references to project outputs were missing data.

#### **IOC-000072008 (Basic MSI, InstallScript MSI)**

The HTML reports that are generated under a release in the Releases view display the Directory table's DefaultDir special values correctly. Previously, the wrong destination was displayed for certain merge modules.

#### **IOC-000072086 (Basic MSI, InstallScript MSI)**

InstallShield no longer crashes if you are debugging a release and the MSI Debugger encounters a property value that exceeds 256 characters.

#### **IOC-000072089 (Basic MSI, InstallScript MSI)**

The Windows Installer property IS\_MAJOR\_UPGRADE is no longer set at run time for a detect-only major upgrade.

#### **IOC-000072145 (InstallScript)**

A cyclic redundancy check (CRC) error no longer occurs during file transfer at run time when a particular large file is included in an installation.

#### **IOC-000072272, IOC-000074450 (Basic MSI)**

When feature prerequisites that are not configured to show progress are installed, the progress bar is now incremented each time one of the feature prerequisites is installed. This occurs on non-Windows XP systems such as Windows Vista, Windows Server 2008, and Windows 2000. On Windows-based systems that use the Aero theme, the progress bar shows animation. Previously on these systems, the progress bar remained empty while the feature prerequisites were installed.

#### **IOC-000072320 (Basic MSI, InstallScript MSI)**

At build time, InstallShield now places all of the .cab files in a single directory with the Setup.exe file and .msi package if the release type is network image, and if the compression type is one .cab file per feature. Previously, the .cab files that contained the first 2 GB of compressed data were placed in Disk1 folder. The remaining .cab files were placed in separate additional folders.

#### **IOC-000072484 (InstallScript, InstallScript MSI)**

If you select the "Select only available languages" check box when you are modifying the values for the Setup Languages setting in the General Information view, the Hebrew and Arabic languages are no longer listed.

#### **IOC-000072596 (InstallScript MSI)**

Folders that were created using the CreateDir function are now removed during uninstallation.

#### **IOC-000072617 (ClickOnce Deployment)**

If you build a ClickOnce Deployment release that fails and then open the Application Information page, InstallShield no longer crashes.

#### **IOC-000072621**

If you try to launch the installation of the InstallScript Object for .NET while the Japanese version of InstallShield is open, the error message that is displayed to inform you that InstallShield must be closed is now displayed in Japanese. Previously, the error message contained garbled characters on a Japanese system.

#### **IOC-000072722, IOC-000075593 (Basic MSI, InstallScript MSI)**

If you add multiple files to a Windows Mobile installation in the Mobile Devices view and select specific target platforms for those files, build error -6516 no longer occurs.

#### **IOC-000072913 (Basic MSI, InstallScript MSI)**

Build error -6151 no longer occurs if the project contains a SQL script file name that contains more than 46 characters.

#### **IOC-000072934 (InstallScript)**

If you view the strings in a project on the Installation Localization page in the Project Assistant or in the String Editor view (which was previously available as string tables in the General Information view) InstallShield does not crash. Previously, InstallShield crashed if it was installed to a short non-default path.

#### **IOC-000073058 (InstallShield MSI Diff)**

If you select the Compare To command on the File menu in InstallShield MSI Diff, the Open dialog box opens. If you select the Windows Installer Databases filter on this dialog box, it now shows .msm and .pcp files, as well as .msi files. Previously, it showed only .msi files.

#### **IOC-000073059 (InstallShield MSI Diff)**

When you are reviewing the differences between two files in InstallShield MSI Diff, InstallShield MSI Diff now displays a pane that shows the old value and the new value of the selected table field, which makes it easy to compare how an individual field was changed. Previously, the differences were displayed only with a tooltip, which disappeared after a few seconds.

#### **IOC-000073194 (Basic MSI, InstallScript MSI)**

Setup.exe no longer checks the Windows Installer version on the target system if you selected "Yes (no Windows Installer engine included)" for the Setup Launcher setting on the Setup.exe tab in the Releases view. Previously in some cases, a run-time error may have been displayed, indicating that a Windows Installer redistributable file could not be found. This may have occurred if the following were true: you selected "Yes (no Windows Installer engine included)" for the Setup Launcher setting, you included an InstallShield prerequisite for Windows Installer 4.5, and the target system did not have Windows Installer 3.1.

#### **IOC-000073301 (Basic MSI, InstallScript MSI)**

When Setup.exe is run in administrative installation mode, the installation no longer checks to see if a newer version of the product is already installed. This enables you to use the /a command-line parameter with Setup.exe to perform an administrative installation of a product when a newer version of that product is already installed. Previously, the administrative installation was not performed.

#### **IOC-000073333 (Basic MSI, InstallScript MSI)**

If you select Yes for the COM Extract at Build setting of two components and COM extraction succeeds for one component but not the other, build warning -4354 occurs. Previously, no build warning was displayed to inform you about the failed COM extraction.

#### **IOC-000073387 (InstallScript, InstallScript MSI)**

The SelectDirEx dialog no longer displays a help button on Windows Vista systems. Previously, this button was displayed, but it did not provide any help messages.

#### **IOC-000073460 (InstallScript)**

If you include a 2-GB-or-larger file in a project as a support file, run-time error -5006 no longer occurs.

#### **IOC-000073606 (Basic MSI, InstallScript MSI)**

If InstallShield extracts COM data from a COM server, HKEY\_CLASSES\_ROOT\Interface registry keys are now created at run time on the target system as needed. Previously, these keys were not created when InstallShield used the Registry table to store the COM data.

#### **IOC-000073614**

Winsock-related registry keys have been added to the registry exclusion list in the Filters.xml file. Therefore, InstallShield no longer automatically adds Winsock-related registry keys to projects when COM extraction is performed.

#### **IOC-000073702 (Basic MSI, InstallScript, InstallScript MSI)**

If you configure an IIS Web site or virtual directory in your project to use the basic authentication method for collecting user name and password information, it is now enabled at run time on Windows Server 2008 systems with IIS 7. Previously, it was not enabled at run time. Note that the basic authentication module needs to be installed on the target system; it is not installed by default.

#### **IOC-000073808, IOC-000075666 (InstallScript)**

The SdLogonUserInformation dialog no longer displays a "server not found" error when the installation is run on a Windows Server 2008 machine that is on the selected domain.

In addition, this dialog now allows the end user to enter user credentials for the local machine.

#### **IOC-000073911 (Merge Module)**

If you use a UTF-8 database for a merge module and for the Basic MSI project that consumes the merge module, you can now use double-byte characters for names of files and directories that will be installed at run time.

#### **IOC-000073989**

If you enter license server information on the Specify License Server panel of the activation wizard and then click the Test Connection link, the activation wizard now properly tests the server connection. Previously, the test always indicated a failed connection, even if it was successful.

#### **IOC-000074105**

The names of the assemblies from the .NET Framework 3.5 have been added to the dependency exclusion list in the Filters.xml file. Therefore, InstallShield no longer automatically adds these assemblies to projects as dependencies when components are scanned for .NET dependencies.

#### **IOC-000074207 (InstallScript MSI)**

InstallScript MSI installations now log event-driven InstallScript functions for uninstallation. In InstallShield 2009, the functions were not logged for uninstallation. Therefore, if the script contained code for creating directories and copying files, those directories and files were not removed during uninstallation.

#### **IOC-000074114 (Basic MSI, InstallScript, InstallScript MSI)**

If you use the XML File Changes view to configure an XML element to be removed when the component is uninstalled, and if the element uses a namespace, the element is now successfully removed at run time. Previously in this scenario, the element was not removed.

#### **IOC-000074208 (Basic MSI, InstallScript MSI)**

When InstallShield builds a release that has a compression type of one .cab file per feature and the project has a very large number of files, the names of the .cab files in the Media table match the ones that InstallShield builds. Previously in one case, the names did not match.

#### **IOC-000074240 (Basic MSI, InstallScript MSI)**

Windows Mobile devices no longer need to be uncradled and then cradled in order to transfer files from the Windows XP system to the device. If the device is connected, the files are transferred right away at run time.

#### **IOC-000074262 (Basic MSI, InstallScript MSI)**

InstallShield uses a newer version of CabWiz.exe for creating mobile device installations. With this version, the maximum binary registry value is now 4,083 bytes, instead of 239 bytes.

#### **IOC-000074291 (Basic MSI, InstallScript, InstallScript MSI)**

If you use the XML File Changes view to configure changes for child elements, the installation no longer adds extra parent elements for those child elements. In addition, if the "Always create this element if it does not already exist" check box is cleared for an element that is not present in the target file, its child elements are also not created.

#### **IOC-000074265, IOC-000077104 (InstallScript, InstallScript MSI)**

If you are using dialog skins and you include a call to Enable (STATUSEX) in your script, the installation no longer crashes. Previously, the installation crashed under certain circumstances.

#### **IOC-000074326**

InstallShield no longer crashes when you use the Windows toolbar to switch between different InstallScript files (.rul).

#### **IOC-000074405, IOC-000075522, IOC-000077042**

The conditions for the Microsoft .NET Framework 2.0 SP1 prerequisite have been corrected. In addition, if the Windows Installer needs to be installed with the installation, it is now installed before the .NET Framework 2.0 SP1 is installed.

#### **IOC-000074455 (InstallScript MSI)**

Installations that are built with InstallShield no longer crash at run time with an unhandled Win32 exception. Previously, this occurred when a command line that was passed to MsiInstallProduct was extremely long, which can occur in projects that use thousands of directories and properties.

#### **IOC-000074520 (Basic MSI, InstallScript MSI, Merge Module)**

InstallShield no longer crashes when you are using the Component Wizard to create a custom-created component and you select a private directory identifier.

#### **IOC-000074525 (InstallScript)**

You can now specify an .spc file and a .pvk file to digitally sign your Setup.exe file but leave the Certificate URL setting blank. Previously, build warning -5032 occurred in this scenario, and the release was not signed.

#### **IOC-000074548 (Basic MSI, InstallScript MSI, Merge Module, Transform)**

If you use the System Search Wizard to create a launch condition and then you later delete the system search, InstallShield also deletes the launch condition. In addition, InstallShield displays a message box that asks you whether you want to delete the associated string entry and conditions. Previously, it was necessary to manually delete the launch condition, associated string entries, and conditions after deleting a system search.

### **IOC-000074664 (Basic MSI)**

The EulaScrollWatcher.dll file that is included with InstallShield now includes support for Unicode characters on run-time dialogs. You can include this file as a custom action in your project if you want to disable the Next button on the LicenseAgreement dialog until the end user reaches the end of the End-User License Agreement (EULA) text in the scrollable EULA control.

Previously, when an end user reached the end of the EULA, the Next button was not enabled if it contained Unicode characters that were not present on the current ANSI Code Page.

### **IOC-000074739 (InstallScript MSI)**

The InstallScript portion of an installation is no longer removed from a target system at run time for InstallScript MSI projects that use detect-only major upgrade items.

### **IOC-000074841**

When you use the Import REG File Wizard to import a .reg file into your project, the registry keys are now displayed in alphabetical order. Previously, the keys were not imported in any particular order. To sort the child keys of any key in the Registry view, select the parent key and then press CTRL+F4.

### **IOC-000074860 (InstallScript)**

InstallScript One-Click Install installations no longer fail if Yes is selected for the Compressed Script setting on the Build tab in the Releases view.

### **IOC-000074866 (Automation Interface)**

If you use CScript to execute multiple VBScript files that use the automation interface to edit InstallShield projects, CScript no longer fails with a memory access violation and a result code of -1073741819.

### **IOC-000074930 (Basic MSI)**

If your release includes a feature prerequisite and you test the user interface of the release from within InstallShield (by clicking the Test User Interface button or by clicking the Test command on the Build menu), the feature prerequisite is no longer launched.

### **IOC-000074931 (Basic MSI, InstallScript MSI)**

When you have specified release flags and product configuration flags for a release that you are building in the Releases view, InstallShield now properly includes and excludes InstallShield prerequisites that have release flags. That is, if you assign a release flag to an InstallShield prerequisite and also specify that flag for the release, for the product configuration, or for both, InstallShield includes that InstallShield prerequisite in the release at build time. Previously, the InstallShield prerequisite was excluded from a release if either set of flags in the Releases view did not match the flag that was assigned to the prerequisite.

### **IOC-000074973**

Renaming an ISWiSQLScript object no longer causes an error on the next call to ISWiSQLConnection.AddSQLScript().

### **IOC-000075060 (Basic MSI, InstallScript, InstallScript MSI)**

The SQLBrowse run-time dialog now shows a list of SQL Servers when the installation is run on a Japanese Windows Vista machine. Previously, the list was empty.

#### **IOC-000075090 (Basic MSI, InstallScript, InstallScript MSI)**

If you build a project that contains an empty SQL script file in the SQL Scripts view, InstallShield no longer displays build error -7086 (An error occurred while encrypting the SQL Script file.).

#### **IOC-000075109 (Basic MSI, InstallScript MSI)**

InstallShield can now extract COM data from a file when left and right brackets ([]) are included in the source path for that file.

#### **IOC-000075161, IOC-000076965 (Basic MSI, InstallScript MSI)**

The .NET Framework 2.0 Chinese (Simplified) language pack is now listed in the Release Wizard.

#### **IOC-000075193 (Basic MSI, InstallScript MSI)**

If one of the advertise options is selected for the Advertise If Prerequisites Are Elevated setting in the Releases view and the installation is run on a Windows XP system, the .msi file is no longer advertised. The advertisement occurs only on Windows Vista or later, where advertisement may help to minimize the number of UAC prompts.

#### **IOC-000075218 (Basic MSI, InstallScript MSI)**

When you build a DVD type of release with custom compression of one .cab file per feature, features are no longer split into multiple .cab files if the DVD capacity has not yet been reached.

#### **IOC-000075289 (InstallScript, InstallScript MSI)**

If you use a dialog skin and select False for the Visible property of an icon control on one of the dialogs, the icon control is now hidden at run time.

#### **IOC-000075360 (InstallScript)**

InstallShield no longer crashes if you build an InstallScript project that contains the InstallScript Object for Crystal Reports.

#### **IOC-000075408 (InstallScript)**

A Korean translation of the message "Another instance of this setup is already running" is now displayed when appropriate. Previously, when the installation displayed the message, it displayed garbled characters.

#### **IOC-000075478**

In the Japanese version of the InstallShield Prerequisite Editor, all of the condition options are now localized into Japanese. Previously, the option called "A registry entry has a specified version value" was displayed in English.

#### **IOC-000075521 (Basic MSI)**

The instance selection dialog in a multiple-instance installation now includes a column for the version number of each existing instance.

#### **IOC-000075729 (Smart Device)**

InstallShield now generates a .cab file at build time for a Smart Device project that includes two files with the same name but different source locations. Previously in this case, InstallShield did not build the .cab file or display any build error.

#### **IOC-000075730 (Basic MSI)**

If you build a release that includes multiple-instance support and the Releases view contains multiple product configurations, the resulting release contains multiple-instance transforms for only the release's product configuration. Previously, the release contained multiple-instance transforms for all of the product configurations in the project.

#### **IOC-000075739 (QuickPatch)**

Nonstreamlined cumulative QuickPatch packages for base packages that include conditions on some components now update these components correctly when intermediate patches are not applied. Note that the intermediate packages must have also been built with this fix in place.

#### **IOC-000075759**

If you type a space as the last character in the file name of a new project in the New Project dialog box, InstallShield now removes that space when creating the InstallShield project file and the release folder. Previously, InstallShield did not remove the space, and this caused some issues.

#### **IOC-000075849 (InstallScript MSI)**

If a prompt-for-original-source dialog needs to be displayed on a target system for an InstallScript MSI installation, it is now displayed. Previously, a retry/cancel message box was displayed instead; clicking Retry displayed the message box again, and clicking Cancel resulted in 1706 and 1603 errors.

Note that the InstallScript event handler OnResolveSource has been removed. The Windows Installer now handles all source resolution. If you added the OnResolveSource event to an InstallScript MSI project in InstallShield 2009 or earlier and then you upgrade that project to InstallShield 2010, that event will no longer be called.

Also note that during a silent installation, there is no way to display the prompt, so the installation is aborted if the original source is needed but it cannot be found.

#### **IOC-000075974 (Basic MSI, QuickPatch)**

When an end user clicks the Update button on the PatchWelcome dialog, the patch now proceeds with the rest of the patch, instead of showing the PatchWelcome dialog again. Previously, Update.exe was launched in an endless loop after each click of the Update button if the original installation included one or more feature prerequisites.

#### **IOC-000075990 (Basic MSI, InstallScript MSI)**

The SQLBrowse dialog no longer displays (local) on machines where only (local)\SQLEXPRESS exists.

#### **IOC-000076083 (Basic MSI)**

If you use a property to let end users change the name of an instance being installed through a multiple-instance installation, Setup.exe now attempts to retrieve the updated product name whenever Setup.exe is relaunched. If the updated product name can be retrieved, it is now included on the instance selection dialog. Previously, the original product name was always displayed.

#### **IOC-000076229 (InstallScript MSI)**

If you use the AddFolderIcon function to create a shortcut, the shortcut is now removed during uninstallation.

#### **IOC-000076257, IOC-000079161 (Basic MSI, InstallScript MSI)**

If you specify a bitmap file with an .ico file extension for the Display Icon setting in the General Information view, error -3204 no longer occurs at build time.

#### **IOC-000076307 (InstallScript, InstallScript MSI)**

If you edit the size of an English dialog in your project on a Japanese system, the dialog is now displayed correctly at run time. Previously in some cases, the watermark or the line next to the watermark was not displayed at run time.

#### **IOC-000076368 (InstallScript, InstallScript MSI)**

If you edit a dialog without changing the dialog size, and the contains double-byte characters, InstallShield no longer changes the size of the dialog.

#### **IOC-000076435 (Transform)**

A response transform now sets the schema value to that of the source .msi package's value. Previously, response transforms changed the schema value to the version of MsiExec.exe on the target system.

#### **IOC-000076499 (Basic MSI, InstallScript MSI)**

If your project includes dynamically linked font files, and you specify your earlier Windows Installer package on the Advanced Settings panel in the Release Wizard, InstallShield now uses upgrade and patch optimization for the font files at build time. That is, InstallShield makes the Font table keys consistent across the releases.

#### **IOC-000076479 (InstallScript, InstallScript MSI)**

If you edit Korean dialogs on a non-Korean system, their size is now correct. Previously in this scenario, the dimensions of the dialogs were changed.

#### **IOC-000076640 (Basic MSI, InstallScript MSI)**

The Crystal Reports 8.5 SP3 object now includes the most recent files.

#### **IOC-000076745 (Basic MSI, InstallScript MSI)**

If you create a condition for a SQL script file in the SQL Scripts view and you select the Run Script During Login check box for that SQL script, the SQL script is no longer executed during login at run time if the condition that you created is not met on the target system.

#### **IOC-000076940 (Basic MSI, InstallScript MSI)**

The Danish translation of the IDS\_\_IsExitDialog\_InstallSuccess string entry has been corrected.

#### **IOC-000076964 (Basic MSI)**

The MSI Debugger no longer encounters an error regarding a corrupted stack when debugging a managed-code custom action.

#### **IOC-000076437 (Basic MSI, InstallScript MSI)**

If you add the predefined search for the .NET Framework 2.0 SP1 to your project through the System Search view or through the Installation Requirements page of the Project Assistant, the installation now checks for the presence of the .NET Framework 2.0 or .NET Framework 2.0 SP1 at run time. Previously, the predefined search's condition was not met if SP1 was installed on the target system.

#### **IOC-000077270**

The automation interface now suppresses interactive error messages.

#### **IOC-000077687**

The value for string 1102 in the 0x041b.ini file now contains the correct Slovak translation.

#### **IOC-000077761 (Basic MSI, InstallScript MSI)**

Windows Installer properties that are associated with SQL connections are now resolved correctly during maintenance and uninstallation when they are used in text replacements that were configured in the SQL Scripts view.

#### **IOC-000078023**

The InstallScript function LaunchApplication no longer consumes all of the CPU time; in addition, the timeout value that is passed to WaitForApplication now waits for all processes in the timeout period that is specified for the nTimeOut parameter. Previously, the CPU process issue occurred after the parent process terminated, but before the child process terminated. It also occurred after the timeout period ended, but both the parent and child processes were still running.

#### **IOC-000078036 (Basic MSI, InstallScript MSI)**

The UI level for a chained .msi package is now limited to be no higher than that of the parent package's current UI level. For example, in the following scenario, the chained .msi package is launched silently: you add a chained .msi package to your project in the Releases view and select Full UI (/qf) for its UI level setting, but the main installation is launched silently (/qn). Previously, the chained package showed exactly the UI level that was authored in the Releases view of the main installation; to restore this behavior, set the property ISChainExceedUILevel equal to the value 1.

#### **IOC-000078272 (Basic MSI, InstallScript MSI)**

The \_Validation table now uses Assembly instead of Assembly\_ for a column name in the MsiPatchOldAssemblyName table.

#### **IOC-000078359 (Basic MSI, InstallScript MSI)**

If the name of the chained .msi package item that you add to the Chained .msi Package area in the Releases view contains more than 9 characters, build error 6151 (Cannot save target database) no longer occurs.

#### **IOC-000078500 (Basic MSI, InstallScript MSI)**

If you use the Internet Information Services view to configure an IIS Web site and you set the Windows Installer property IISREFERLEGACYOBJECTS equal to 1, the Web site can now be successfully installed and uninstalled on a target system that has the IIS 6 Metabase Compatibility feature installed. Previously in some cases, the Web site could not be installed, and a rollback left the corrupt Web site on the target system.

#### **IOC-000078631 (Basic MSI)**

If Unicode is selected for the Setup Launcher Type setting, the Setup.ini file that is created at build time is now Unicode. Previously, Unicode was not used for Setup.ini, and run-time error 1155 occurred under certain conditions.

#### **IOC-000078660 (Basic MSI)**

The ShowMsiLog custom action now launches Notepad.exe from the SystemFolder directory, instead of from the WindowsFolder directory. Thus, if your installation is run on Windows Vista or later and the end user indicates on the SetupCompleteSuccess dialog that they want to view the log file, the installation launches Notepad.exe from the SystemFolder directory. This change was made because on Windows Server 2008 Standard Edition, Notepad.exe is available in the System32 directory, but not the Windows directory. Note that behavior is available by default in all new Basic MSI projects that are created in InstallShield 2010.

#### **IOC-000078664 (Merge Module)**

The \_Validation table now has the correct ServiceInstall table value for the MaxValue column. Previously, the wrong value was used, and validation error ICE03 occurred in some cases.

#### **IOC-000078792**

When you use the Import REG File Wizard to import a .reg file into a project on a Japanese system, all of the registry keys are now imported.

#### **IOC-000078795 (Basic MSI, InstallScript MSI)**

If you upgrade an InstallShield 12 project that has the Update Service merge module to InstallShield 2010, error - 4075 (File not found) does not occur at build time.

#### **IOC-000078864 (InstallScript, InstallScript MSI)**

When the installation displays the SQLServerSelect dialog at run time, it now use the database server that is identified for the szServer parameter of the SQLServerSelect function as the default value. Previously, the function always displayed the first item of existing SQL servers as the default value.

#### **IOC-000079127 (Basic MSI, InstallScript MSI)**

If you use the CmdLine column of the IsSelfReg table to specify an installation parameter and an uninstallation parameter to be passed to your self-registering COM server when it is being registered at run time, the correct parameters are now used at run time during installation and uninstallation. Previously, the installation parameter was passed at installation and uninstallation.

#### **IOC-000079131 (Basic MSI, InstallScript MSI)**

If you use MSBuild to build an InstallShield project and you set a path variable override in the .isproj file, the path variable is now overridden with the new value that you specify.

#### **IOC-000079460 (Basic MSI, InstallScript MSI)**

If you select one of the advertise options for the Advertise If Prerequisites Are Elevated setting on the Setup.exe tab for a release in the Releases view, and if the end user selects a custom INSTALLDIR location at run time, the product is now installed to the custom location. Previously, the product was installed to the default INSTALLDIR value unless INSTALLDIR had been added to the SecureCustomProperties property.

#### **IOC-000080958 (Basic MSI, InstallScript MSI)**

If you specify a digital certificate for signing your .msi package, InstallShield now uses that digital certificate to sign any language transforms (.mst files) for your release. Therefore, a language transform no longer results in an unknown publisher UAC warning on Windows 7 systems.

## **System Requirements**

This section contains the minimum requirements for systems that run InstallShield (the authoring environment), as well as for target systems that run the installations created with InstallShield (the run-time environment).

### ***For Systems Running InstallShield***

#### **Processor**

Pentium III-class PC (500 MHz or higher recommended)

#### **RAM**

256 MB of RAM (512 MB preferred)

#### **Hard Disk**

500 MB free space

## Display

Designed for XGA resolution at 1024 × 768 or higher

## Operating System

Windows XP  
Windows Server 2003  
Windows Vista  
Windows Server 2008  
Windows 7  
Windows Server 2008 R2

## Browser

Microsoft Internet Explorer 5.01 (IE 5.5 or later recommended)

## Privileges

Administrative privileges on the system

## Mouse

Microsoft IntelliMouse or other compatible pointing device

## *For Target Systems (Desktop Computers)*

Target systems must meet the following minimum operating system requirement:

Windows 2000  
Windows XP  
Windows Server 2003  
Windows Vista  
Windows Server 2008  
Windows 7  
Windows Server 2008 R2

## *For Target Systems (Mobile Devices)*

InstallShield includes support for adding mobile device installations to desktop installations that use Microsoft Windows Mobile Device Center or Microsoft ActiveSync to transfer files to a mobile device.

InstallShield also includes support for straight-to-device installations that do not use Windows Mobile Device Center, ActiveSync, or any other desktop component.

For an overview of the different options that InstallShield supports, see "Creating Installations for Mobile Devices" in the InstallShield Help Library.

## Windows Mobile Device Requirements

InstallShield supports many Windows Mobile platforms and processors. The Windows Mobile platforms are:

- Windows Mobile 6.x Professional and Classic
- Windows Mobile 6.x Standard
- Windows Embedded CE 6.x
- Windows Mobile 5.0 for Pocket PC
- Windows Mobile 5.0 for Smartphone

- Windows CE .NET 5.0
- Windows CE .NET 4.x
- Pocket PC 2003
- Pocket PC 2002
- Pocket PC
- Palm-size PC 2.11
- Palm-size PC 2.01
- Handheld PC 2000
- Handheld PC Pro
- Handheld PC 2.0
- Smartphone 2003
- Smartphone 2002

Note that if a platform is not included in the list, it does not mean InstallShield does not support it. It simply means that you cannot set conditions for that specific platform by default. To add support for additional platforms or to change the conditions for targeting a specific platform, you can modify the Settings.xml file that is installed with InstallShield. For more information, see "Modifying the List of Available Windows Mobile Platforms or their Associated Settings" in the InstallShield Help Library.

InstallShield includes support for the following Windows Mobile processors:

- ARM920
- ARM820
- ARM720
- Common Executable Format
- Hitachi SH4
- Hitachi SH3E
- Hitachi SH3
- i686
- i586
- i486
- MIPS R4000
- MIPS R3000
- MIPS R2000
- SHx SH4
- SHx SH3
- StrongARM-XScale

### **Palm OS Device Requirements**

InstallShield supports Palm OS 3.5 and later.

### **Desktop Requirements for Windows Mobile Device Installations**

Requirements for the desktop computers that are used to install applications on Windows Mobile devices are:

- Microsoft ActiveSync 3.x or later on Windows XP (ActiveSync 4.x is required for Windows Mobile 5.x or later devices)
- Microsoft Windows Mobile Device Center on Windows Vista
- Administrative privileges

### **Desktop Requirements for Palm OS Device Installations**

Palm HotSync is required for the desktop computers that are used to install applications on Palm OS devices.

### **Known Issues**

For a list of known issues, see Knowledge Base article [Q200331](#).