InstallShield 2011

Release Notes

originally released August 2010; updated to include Hotfix A, released October 2010

Introduction

InstallShield is the industry standard for authoring high-quality Windows Installer– and InstallScript-based installations, as well as Microsoft App-V packages. InstallShield 2011 offers many new features and enhancements that make it easy to use the latest technologies, improve the quality and reliability of your products, increase your productivity, and collaborate with other team members.

With InstallShield 2011, InstallScript installations and InstallScript custom actions support Unicode, a necessity for developing modern multilanguage installations. If you use Team Foundation Server, you can take advantage of the new tight integration with InstallShield as you develop, maintain, build, test, and deploy your products and their installations. Now you can specify custom source locations for InstallShield prerequisites; this flexibility enables you to easily add to your projects InstallShield prerequisites that are stored in source code control and to share a common set of prerequisites across multiple machines. InstallShield also lets you emphasize your brand with customizable Setup.exe properties and icons, streamline your build process by specifying commands to be run automatically during various stages of builds, and increase your efficiency with improved script editors. Several other new advancements make it easier for you to create 64-bit installations or 64-bit App-V packages.

InstallShield includes a command-line Standalone Build tool that enables you to install on build machines only the part of InstallShield that builds the installations, plus any redistributables that you want to include. The Standalone Build is now included with the Professional and Premier editions of InstallShield. If you need extra licenses of the Standalone Build, you can also now purchase them separately.

Hotfix A for InstallShield 2011 lets you add to Basic MSI and Merge Module projects any DIM files that contain shortcut sets and shortcuts that were added in InstallShield Collaboration. It also includes additional changes.

For the latest information about InstallShield 2011, including updates to these release notes, see Knowledge Base article Q208910.

Changes in Hotfix A (October 2010)

To obtain Hotfix A, see KB article Q210896.

Hotfix A for InstallShield 2011 lets you add to Basic MSI and Merge Module projects any DIM files that contain shortcut sets and shortcuts that were added in InstallShield Collaboration. It also includes the following additional changes.

Digitally Signing with a .pfx File

If you specify a .pfx file and a certificate password on the Signing tab for a release in the Releases view, InstallShield can now build a release. Previously in some cases, InstallShield did not save the password in the project file correctly, so the build failed with error -1027.

1-JQWZA (Basic MSI, InstallScript MSI, Merge Module)

The Component Wizard now lets you create a font type of component for an .otf file that is not installed on your machine. Previously, the only way to specify an .otf file in this wizard was to select it from a list of the font files that were installed on your machine.

IOA-000053017 (Basic MSI, InstallScript, InstallScript MSI, MSI Database, Transform)

InstallShield no longer crashes if you are using the XML File Changes view to test installation changes to an XML file that has an element name or attribute value with special characters.

IOA-000056234 (InstallScript)

An InstallScript One-Click Install installation that is digitally signed with a non-test certificate no longer shows the security warning (that is, the warning that is displayed if the installation was not signed); it now shows the appropriate digital certificate information. Previously in some cases with digital certificates from some vendors, the digital certificate was not recognized, and the security warning was displayed.

IOA-000057561 (Basic MSI, Merge Module)

If you set a value in the Version property of a file that you added to a DIM, the file's version is now updated as expected in the InstallShield project that consumes that DIM; that is, InstallShield uses the value that you specify in the Version field of the File table of the .msi or .msm file. Previously, setting a value in the Version property of a file in a DIM had no effect.

IOA-000057579 (Basic MSI, InstallScript, InstallScript MSI)

If you have an InstallShield prerequisite in your project and the InstallShield prerequisite has a registry condition, the installation now properly checks 64-bit Itanium systems to verify whether that condition is met. Previously, the registry check did not occur on 64-bit Itanium systems, so the installation attempted to install the prerequisite, even if it was already present.

IOA-000057879 (Basic MSI, InstallScript MSI, Merge Module)

If you upgrade a project from InstallShield 2009 or InstallShield 2010 to InstallShield 2011, and if the original project had a managed-code custom action that was configured to be installed with your product or located in a path that was defined with a property, the resulting installation no longer fails when that managed-code custom action is launched.

IOA-000057885

A catastrophic failure error no longer occurs in the following scenario: You add a Visual C# .NET project to a solution, you add an InstallShield project to the solution, and then you click various places in the Visual C# .NET project.

IOA-000058043 (InstallScript)

If a shared component that has no files is associated with more than one feature and an end user deselects one of those features at run time, a catastrophic failure error no longer occurs during file transfer.

IOA-000058053, IOA-000058216

When passing a BYVAL WSTRING to a DLL function call, the InstallScript engine no longer truncates data after the first null character; it now passes all the data stored in the string. This resolves problems with calls to APIs that use strings with embedded nulls and/or double-null terminated strings (including the InstallScript functions AddProfString and StrPutTokens).

IOA-000058256 (Basic MSI)

If a feature prerequisite references the Windows Installer property ISPREREQDIR and the installation is not running on a Windows XP system, the ISPREREQDIR property is now resolved properly. Previously in this case, ISPREREQDIR was resolved on Windows XP systems, but not on Windows 2000 systems or on Windows Vista and later systems.

IOB-000058706 (Basic MSI, Merge Module)

If you import a DIM file that contains a file set or registry set with a custom target platform condition, and the custom target platform condition specifies a specific architecture such as amd64, InstallShield now sets the condition correctly in the .msi or .msm file that it creates.

IOC-000069399 (Basic MSI, Merge Module)

If you import a DIM file that contains a case-insensitive string comparison, InstallShield now sets the condition correctly in the .msi or .msm file that it creates. For example, if you create a DIM that contains a registry set with a condition such as **MYPROPERTY~="value"** and include that DIM in a Basic MSI or Merge Module project, InstallShield adds that condition to the registry data in the .msi or .msm file that it creates at build time. Previously, InstallShield used a condition such as **MYPROPERTY="value"**.

New Features in InstallShield 2011 Original Release Version (August 2010)

Unicode Support in InstallScript

InstallShield now lets you create InstallScript installations and InstallScript custom actions that support the use of Unicode in run-time strings, files, paths, registry entries, and other installation data. In addition, the InstallScript compiler and the InstallScript engine now let you pass pointers to Unicode strings to functions that are implemented outside script code; they also let you store Unicode strings in structures that are passed outside script code. Other changes in InstallScript offer additional benefits for fully supporting modern multilanguage installations.

As a result of these changes, any language displays correctly on a system that has support for it installed. End users no longer need to match the language that is used on their systems for non-Unicode programs with the language that is used in the installation. Note that the font must be installed on the target system. On some versions of Windows, the fonts for some languages are not installed by default. For example, Japanese fonts are not installed on a Windows XP English system by default; in order for an installation to use Japanese characters on such a system, the fonts would need to be installed.

Unicode Setup Launchers for InstallScript and InstallScript MSI Installations

Now all Setup.exe and Update.exe files that are built in InstallShield are Unicode. Previously, all Setup.exe and Update.exe files that were built in InstallScript and InstallScript MSI projects were ANSI.

A Unicode setup launcher can correctly display characters in the user interface of the setup launcher, regardless of whether the target system is running the appropriate code page for the language. An ANSI setup launcher correctly displayed characters in the setup launcher dialogs if the target system was running the appropriate code page.

However, it may have displayed garbled characters in those dialogs if the target system was not running the appropriate code page.

End users can launch Setup.exe and Update.exe files from within Unicode paths, regardless of the language on the target system. For example, end users can now launch the installation in the following path on an English system: C:\Users*Japanese characters*\Desktop\Setup.exe. Previously, the installation would fail.

Unicode Support for Files, Folders, Registry Entries, and Support Files

Unicode support has been added to key parts of the installation run time, enabling you to use characters from any languages simultaneously in file names, folder names, registry entries, and support file names. This enables you to install, for example, a file that has Japanese characters in its name or target path on an English system. Mixing languages works correctly regardless of the current language of the target system.

Pointer Support

The InstallScript engine and compiler now support a new pointer type called WPOINTER; wpointer and LPWSTR are equivalent names that are also available. Thus, for example, if you have a DLL function that accepts pointers to Unicode strings in its parameters, you can use this new pointer type; when the DLL function is called in the script at run time, the InstallScript engine passes a pointer to a Unicode copy of the strings instead of an ANSI version. Previously, it was possible to pass a pointer to only an ANSI copy of the strings.

Note that passing Unicode strings with the WSTRING data type that was introduced in an earlier version of InstallShield is still supported.

Structure Support

Structures in InstallScript can contain any basic data type, including strings and pointers, or other structures. Now if a structure needs to contain a Unicode string and the structure is passed to an external DLL, the InstallScript engine can distinguish between string member types in that structure, and then size the structure and calculate member offsets correctly. String members that need to be stored and passed as Unicode can be declared with the WSTRING type.

Previously, if a structure needed to contain a Unicode string and the structure was passed to an external DLL, the InstallScript engine assumed that any strings in the structure were ANSI. As a result, the size of the structure and member offsets in the structure could have been wrong, causing the DLL to incorrectly read or write data related to the structure. Attempting to use WSTRING for string members of a structure did not have any effect.

You can leave existing strings as STRING types; the InstallScript engine will continue to treat these as ANSI strings when passing them outside of script code.

Pointer members in structures can also now be declared as WPOINTER. This enables you to store pointers to Unicode strings in a structure.

String Table Support

InstallShield now uses Unicode encoding to build the string tables for InstallScript projects. Because of this support, string tables for InstallScript projects can now contain multiple languages, and they are not affected by a target system's code page settings. In addition, string tables can now include strings for languages—such as Hindi—that have no code page.

It is recommended that any strings that will be displayed in the user interface of an installation be stored in the string table. Although strings can be written directly in InstallScript code, they are not stored as Unicode; thus,

they are displayed correctly only when the installation is run on systems that have the correct code page. Storing strings in the string table and referencing the string identifiers in InstallScript code eliminates this issue.

This feature resolves the following issues: 1-14SJA9, 1-17O0WL, IOA-000051919, IOA-000052767, IOA-000054177, IOB-000050617, IOB-000058234, IOC-000050725, IOC-000063585.

Unicode InstallScript Dialogs

InstallScript dialogs now include support for Unicode. This enables you to mix, for example, Japanese and German, or Russian and Polish, on InstallScript dialogs. These mixed languages work correctly regardless of the current language of the target system.

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

This feature resolves issue IOC-000057769.

Unicode Support in the InstallScript Debugger

The InstallScript Debugger now has support for Unicode. Thus, for example, if you are debugging a line of InstallScript code such as szMsg = @ID_MSG, you can now see the value in szMsg, regardless of what language operating system you are using and what language is used for the strings. If the value of ID_MSG contains Chinese characters or characters from mixed languages, you can now see the appropriate characters instead of question marks in various areas of the InstallScript Debugger: in the variable window, in the watch window, and in the tooltip that appears for the variable in the script window.

This feature resolves issue IOB-000059145.

Unicode Support in InstallShield Cabinet and Log File Viewer

The InstallShield Cabinet File Viewer and the InstallShield Log File Viewer have been combined into one new tool—the InstallShield Cabinet and Log File Viewer—which lets you open and review InstallScript cabinet files (.cab), InstallScript header files (.hdr), and InstallScript log files (.ilg). This tool now includes Unicode support. Thus, the tool can now correctly display the characters in each file name, registry key, shortcut, and other data in the .cab, .hdr, or .ilg file, regardless of what language operating system you are using and what language is used for the strings. Previously, in some cases, some of the characters were displayed as question marks.

The cabinet and header file support applies to the following project types: InstallScript and InstallScript Object.

The log file support applies to the following project types: InstallScript and InstallScript MSI.

Integration with Team Foundation Server (TFS)

InstallShield has enhanced support for integrating with Team Foundation Server (TFS) 2010.

Now when you are using InstallShield from within Visual Studio 2010, you can access the Source Control Explorer to integrate your InstallShield project with Team Foundation version control and manage changes to your InstallShield projects and your Visual Studio solutions.

You can also use Team Foundation Build to compile, test, and deploy your InstallShield projects and your Visual Studio solutions on a regular basis, or on demand. Your installation is automatically updated with your latest source files every time that your solution is built.

In addition, if you install Team Explorer on the same machine that has InstallShield and Visual Studio, you can use Team Explorer from within your InstallShield projects that are open in Visual Studio. This enables you to perform tasks such as the following:

- Use Source Control Explorer when you are working on your InstallShield projects.

- Configure builds for your InstallShield projects and Visual Studio solutions.
- Queue new builds.
- Track work items such as bugs and tasks for your InstallShield projects and your Visual Studio solutions.

Microsoft SQL Server 2008 R2 Support

InstallShield now includes support for running SQL script on SQL Server 2008 R2. In addition, InstallShield includes SQL Server 2008 R2 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 is available in the following project types: Basic MSI, InstallScript, and InstallScript MSI.

This feature resolves the following issues: IOA-000055382 and IOA-000055685.

New InstallShield Prerequisites for SQL Server 2008 R2 Express, SQL Server Native Client, Visual C++ 2010, and Other Redistributables

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

- Microsoft SQL Server 2008 R2 Express
- Microsoft SQL Server 2008 Native Client 10.00.2531
- Microsoft SQL Server Native Client 9.00.4035
- Microsoft SQL Server System CLR Types 10.00.2531
- Microsoft SQL Server 2008 Management Objects 10.00.2531
- Microsoft Visual C++ 2010 Redistributable Package
- Microsoft Visual C++ 2008 SP1 Redistributable Package
- Microsoft Visual C++ 2005 SP1 Redistributable Package (x64)
- Windows Installer 3.1 - Japanese
- MSXML 6.0 SP1 - Japanese
- Microsoft .NET Framework 4.0 Client Japanese Language Pack
- Microsoft .NET Framework 4.0 Full Japanese Language Pack

This feature resolves the following issues: IOA-000053708, IOA-000054694, IOC-000058108.

Improved Script Editors for Creating and Modifying InstallScript Code, SQL Scripts, VBScript Custom Actions, and JScript Custom Actions

Several views in InstallShield have an improved script editor that you can use to write code for your projects. The script editors include the following improvements:

- **Expanded auto completion**—When you are typing in a script editor, InstallShield displays a pop-up list of alphabetically ordered functions, keywords, and constants that begin with the letters that you are typing. Instead of manually typing the entire word, you can select it in the pop-up list, and InstallShield adds it to your script.

  If you are using the script editor in the InstallScript view, you can enter the string constant operator (@) to see a pop-up list of the available string identifiers. You can select the appropriate one in the list, and InstallShield adds it to your script.

  If auto completion for local variables is also enabled, the pop-up list in the script editor of the InstallScript view also contains local variables.

Auto completion can increase your efficiency because it can reduce the time that you spend typing code. It can also help you avoid typographical errors in your code.

The script editor in the InstallScript view of earlier versions of InstallShield is the only script editor that provided some auto completion support; thus, auto completion functionality was not available for SQL scripts, VBScript code, or JScript code. In addition, the available pop-up list was limited to InstallScript functions; none of the keywords, constants, local variables, or string identifiers were available for auto completion.

- **Detailed InstallScript Function Call Tips**—If function call tips are enabled, InstallShield shows InstallScript function call tips—a type of tooltip—when you are entering function calls in your script in the InstallScript view. A function call tip shows a built-in function's parameter information. It also shows a description of the built-in function, as well as a description of the parameter that you are entering. The detailed call tips enable you to see help information about a function without switching from the script editor to the help library, and then back to the script editor.

  The InstallScript view in earlier versions of InstallShield included support for call tips, but the call tips did not display as much information. The call tips showed a function call with all of the function's parameters; however, the call tips did not include a description of the function, or of the parameter that you are entering.

- **Syntax folding**—InstallShield now lets you specify whether you want the script editors in various views in InstallShield to include support for syntax folding. When syntax folding is enabled, InstallShield adds a plus sign (+) or a minus sign (–) in the margin next to each line of code that starts an expandable or collapsible block of script. You can click a plus sign to expand hidden code, or a minus sign to hide visible code.

  Syntax folding can help you minimize the clutter of large scripts and focus on the code that is relevant to the work that you are currently doing. It can also help you see the overall structure of a script.

The views that contain the improved script editor are the InstallScript view, the SQL Scripts view, and the Custom Actions and Sequences view (when you are viewing a VBScript or JScript file in this view).

By default, syntax folding is disabled in the script editors; auto completion, local variable auto completion, and function call tips are enabled. To enable or disable any of the aforementioned functionality, use the Script Editor Properties dialog box. This dialog box also lets you modify other settings in the script editors, such as font, syntax colors, and line numbering. To access this dialog box, right-click in a script editor and then click Properties.

Ability to Create 64-Bit Microsoft App-V Packages

The Microsoft App-V Assistant in InstallShield includes support for converting a 64-bit Windows Installer package into a 64-bit Microsoft App-V package, which can be deployed on 64-bit Windows systems with the 64-bit Microsoft App-V 4.6 client. This, in combination with the new 64-bit repackaging support that is available with the Repackager, enables you to convert any 64-bit installation into a 64-bit App-V package.

It is recommended that you perform the conversion of 64-bit Windows Installer packages to App-V packages on a 64-bit Windows system. If you attempt the conversion on a 32-bit system, it could result in a failure to extract COM information for 64-bit binaries. Also, in some cases, Windows Installer packages contain shortcuts that target executable files that are not found in the package itself. If these shortcuts target executable files are found in 64-bit locations, these shortcuts are not handled correctly on 32-bit systems.

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

Ability to Specify a Search Path for InstallShield Prerequisites; Path Variable and Relative Path Support for Source Locations of Prerequisite Files

InstallShield now lets you specify the folders where InstallShield should search for InstallShield prerequisite files (.prq files). This functionality makes it easier for teams of users to share InstallShield prerequisites with each other,

and for storing prerequisites in source code control. Previously, InstallShield searched for .prq files in the following location only: *InstallShield Program Files Folder*\SetupPrerequisites.

InstallShield offers several ways for specifying the folders:

- If you are editing or building from within InstallShield, use the new Prerequisites tab on the Options dialog box—which is displayed when you click Options on the Tools menu—to specify a comma-delimited list of machine-wide folders and current-user folders. This tab is similar to the Merge Modules tab on the Options dialog box, which lets you specify search paths for merge modules.
- If you are building from the command line with ISCmdBld.exe, use the new -prqpath parameter to specify a comma-delimited list of folders.

  If you use an .ini file to specify ISCmdBld.exe parameters, you can use the new PrerequisitePath parameter in the [Mode] section of your .ini file to specify a comma-delimited list of folders.
- If you are building through MSBuild or Team Foundation Server (TFS), use the new PrerequisiteSearchPatch parameter on the InstallShield task. This parameter is exposed as the ItemGroup InstallShieldPrerequisitePath when the default targets file is used. To specify multiple paths, use an ordered array of paths.

When you are using the Files to Include tab in the InstallShield Prerequisite Editor to add files to an InstallShield prerequisite, the editor now uses predefined path variables such as <WindowsFolder> and <ISProductFolder> if appropriate. In addition, if the files that you are adding are stored in the same folder as the InstallShield prerequisite's .prq file—or a subfolder of the folder that contains the .prq file—the InstallShield Prerequisite Editor uses a relative path for the file in the .prq file. Note that if you view the file's path on the Files to Include tab, the InstallShield Prerequisite Editor lists the full path, not the relative path.

Note that if you use the Save As command in the InstallShield Prerequisite Editor to change the location of a .prq file, the InstallShield Prerequisite Editor updates the paths of the prerequisite's files if appropriate.

This feature resolves the following issues: IOA-000055078, IOA-000056459, IOC-000054166, IOC-000065360, IOC-000067484, IOC-000078297.

New Hyperlink Control for Windows Installer–Based Dialogs

The Controls toolbar includes a new hyperlink control that you can use in the Dialogs view for Windows Installer–based dialogs. The hyperlink control displays an HTML link; clicking the link at run time opens a page in the default browser on the target system.

Windows Installer 5 includes support for this new hyperlink control. If this control is used on a dialog that is displayed on a system that has an earlier version of Windows Installer, run-time error 2885 occurs, and the installation aborts. Therefore, if you want to use the hyperlink control on a dialog but your installation targets systems that have Windows Installer 4.5 or earlier, it is recommended that you include two versions of the dialog in your project: one with the hyperlink control, and one without it. Add conditions to the dialogs to show or hide them, depending on the version of Windows Installer that is present.

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

Ability to Specify a Custom Icon and Custom Version Resource Properties for Setup.exe and Update.exe

InstallShield now lets you use custom icon and custom version resource properties for Setup.exe files that you create at build time. The icon and the version resource properties are displayed on the Properties dialog box for Setup.exe; this Properties dialog box opens when end users right-click the Setup.exe file and then click Properties. End users can also see the icon when they view your Setup.exe file in Windows Explorer. This support is available in Basic MSI, InstallScript, and InstallScript MSI projects.

The same functionality (the ability to specify a custom icon and custom version resource properties) is also now available for Update.exe files that you create in Basic MSI, InstallScript MSI, and QuickPatch projects.

Custom Icon for Setup.exe and Update.exe

The Setup.exe tab in the Releases view has a Setup.exe Icon File setting that lets you specify the icon that should be used for your Setup.exe setup launcher. The icon can be in an .exe, .dll, or .ico file. If you do not specify an icon, InstallShield uses a default icon for your Setup.exe file.

Previously, support for specifying a custom Setup.exe icon was available for InstallScript projects if the Setup.exe file that you were building was a self-extracting, single-file setup launcher. The support was not available in Basic MSI or InstallScript MSI projects. In addition, it was not available for uncompressed InstallScript installations.

A new Icon setting lets you specify the icon that should be used for your Update.exe launcher. This setting is available on the Advanced tab for a patch configuration in the Patch Design view of Basic MSI and InstallScript MSI projects. It is also available on the Advanced tab of the Build Settings area in the General Information view of QuickPatch projects. If you do not specify an icon, InstallShield uses a default icon for your Update.exe file.

Previously, InstallShield did not include any support for specifying a custom icon for Update.exe files.

Custom Version Resource Properties for Setup.exe and Update.exe

When InstallShield is configuring the following version resources of your Setup.exe setup launcher at build time, it now uses the custom information that you entered in the General Information view and the Releases view:

- Company name
- Product name
- Product version
- Copyright
- File version
- File description

To use a custom copyright notice and a custom file description, ensure that you select Yes in the Use Custom Version Properties setting for the release in the Releases view.

Previously, InstallShield did not use any custom information in many cases. For example, the InstallScript Setup.exe files that InstallShield previously created contained details that pertained to the version of InstallShield that was used to build the Setup.exe file. Thus, the product name that was displayed in the Setup.exe Properties dialog box was InstallShield, rather than the name of your product.

The ISWiRelease object of the automation interface includes a new LauncherFileDescription property that lets you set the file description of the Setup.exe file. Several already existing properties let you configure the other version resource properties of Setup.exe. The ISWiProject object includes the following properties: CompanyName, ProductName, and ProductVersion. The ISWiRelease object includes the following properties: LauncherCopyright and UseMyVersionInfo (which is the equivalent of the Use Custom Version Properties setting).

Several new settings—Company Name, Product Name, Product Version, Description, and Copyright—let you specify the custom information that you want InstallShield to use when you build an Update.exe file. These settings are available on the Advanced tab for a patch configuration in the Patch Design view of Basic MSI and InstallScript MSI projects. They are also available on the Advanced tab of the Build Settings area in the General Information view of QuickPatch projects.

Previously, InstallShield did not use any custom version resource information for Update.exe files.

This feature resolves the following issues: IOA-000031784, IOA-000035007, IOA-000043150, IOA-000052852, IOA-000056121, IOC-000051102, IOC-000055158, IOC-000070983.

Ability to Specify Commands that Run Before, During, and After Builds

The Premier edition of InstallShield includes new release settings that you can use to specify commands that you want to be run at various stages of the build process. These new settings are available on a new Events tab when you select a release in the Releases view:

- **Prebuild Event**—Use this setting to specify the command that you want to be run before InstallShield starts building the release. This event runs after InstallShield creates the release folder and log file, but before InstallShield starts building the release.

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

- **Precompression Event**—Use this setting to specify the command that you want to be run after InstallShield has built the .msi package and the .cab files (if your product's data files are to be stored in .cab files). Note that this event occurs after .cab files are streamed into the .msi package, but before the .msi package has been digitally signed and streamed into the Setup.exe file.

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

- **Postbuild Event**—Use this setting to specify the command that you want to be run after InstallShield has built and signed the release.

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

The new Events tab replaces the previously available Postbuild tab. Note that the settings that were previously available on the Postbuild tab are now displayed on the new Events tab. In addition, the publish-related settings that were previously available on the Build tab in Merge Module projects have been moved to the Events tab.

The automation interface now includes support for the new build event settings. The ISWiRelease object includes three new properties that let you specify commands for various stages of the build process:

- PrebuildEvent
- PrecompressionEvent
- PostbuildEvent

Ability to Set an Expiration Date for Setup.exe

InstallShield now lets you set an expiration date, as well as an expiration message, for Setup.exe. If end users try to run Setup.exe on or after the date that you have specified in your project, the expiration message is displayed, and the installation exits.

To set an expiration date and a message for your Setup.exe file, use the new Expiration Date and Expiration Message settings on the Setup.exe tab for a selected release in the Releases view. By default, no expiration date is configured for Setup.exe. If you specify an expiration date, you can use the default expiration message, or specify your own custom message.

The automation interface now includes support for these new settings. The ISWiRelease object includes new properties that let you set the expiration date and message: ExpirationDate and ExpirationMessage.

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

Ability to Import Visual Studio Setup and Merge Module Projects into Existing InstallShield Projects; Improvements for the Project Converter

InstallShield now lets you import a Visual Studio setup project (.vdproj) into an InstallShield Basic MSI or Merge Module project (.ism), or import a Visual Studio merge module project (.vdproj) into an InstallShield Basic MSI or Merge Module project (.ism). These tasks enable you to develop InstallShield installation and merge module projects that contain the same data and settings that were in your Visual Studio project. The wizard imports the project outputs, files, registry keys, file extensions, custom actions, target system searches, and launch conditions from your Visual Studio project into your existing InstallShield project.

To import a Visual Studio project into an existing InstallShield project, use the new Visual Studio Deployment Project Import Wizard in InstallShield. The wizard lets you choose whether to import or ignore certain settings in the Visual Studio project.

The existing support for converting a Visual Studio project into a new InstallShield project has been expanded. If your Visual Studio project contains predefined prerequisites, InstallShield now converts them to equivalent InstallShield prerequisites during the conversion process. This same prerequisite conversion functionality is available if you use the new wizard to import a Visual Studio project into an InstallShield project.

If your Visual Studio project contains one or more project outputs, use the import wizard instead of the conversion process. The InstallShield project must be in the same Visual Studio solution that contains the Visual Studio setup or merge module project and all of its project dependencies. Note that you must be using InstallShield from within Visual Studio when you use the import wizard in order for the wizard to import the project outputs into your InstallShield project.

Unicode and UTF-8 Support for SQL Scripts

InstallShield now has design-time and run-time support for SQL scripts that have either Unicode BOM encoding or UTF-8 BOM encoding. You can use the SQL Scripts view to add SQL scripts with either of these encoding types to your project. You can also edit those SQL scripts from within that view as necessary. At run time, the SQL scripts are executed during installation and uninstallation when needed.

Previously, InstallShield had run-time support but not design-time support for SQL scripts with Unicode BOM encoding. Thus, when you added a script with this encoding to the SQL Scripts view of your project, InstallShield prompted you to specify whether you wanted to convert the script to ANSI format. If you allowed InstallShield to convert it to ANSI, you could edit it from within the SQL Scripts view. However, in some cases, it might have displayed and used garbled characters at design time and at run time. If you did not allow InstallShield to convert it to ANSI, the file remained with Unicode BOM encoding. Although this encoding allowed your installation to properly use strings in languages that did not match the code page of the target system at run time, you could not view or edit the script from within InstallShield.

In addition, InstallShield previously did not have adequate support for SQL scripts with UTF-8 BOM encoding. If you added a SQL script with this encoding type to your project and the script contained strings in a language that did not match the code page of the development system, the SQL Scripts view treated the file as an ANSI file, and it may have displayed garbled characters. In addition, garbled characters were used when your SQL script was executed at run time. Also, the byte order mark was represented as garbled characters.

If you create a new SQL script from within the SQL Scripts view, InstallShield uses Unicode BOM encoding for the file. If you prefer to use ANSI or UTF-8 BOM encoding for the SQL scripts view, it is recommended that you create an .sql file with the proper encoding in a different tool, and then import or insert the script in the SQL Scripts view of your project.

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

This feature resolves issue IOC-000066750.

Predefined System Searches for SQL Server 2008 Express SP1 and Adobe Reader 9

InstallShield has new predefined system searches:

- SQL Server 2008 Express SP1
- Adobe Reader 9

If your installation requires one or both 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 feature applies to Basic MSI and InstallScript MSI projects.

64-Bit Support for Repackager

Repackager now has the ability to repackage 64-bit installations. Previously, only 32-bit installations could be repackaged.

The Repackaging Wizard remains a 32-bit application, but you can run it on both 32-bit (x86) and 64-bit (x64) Windows systems.

Note that although you can use Repackager on a 64-bit system to repackage a 32-bit installation that can be run on 64-bit or 32-bit systems, it is recommended that you use Repackager on a 32-bit system to avoid inadvertently capturing any 64-bit data (such as 64-bit directories or registry locations). If Repackager captures any 64-bit data, it flags the package as a 64-bit installation, meaning that it can be run only on 64-bit systems.

Repackager is available in the Premier edition of InstallShield. It is also included with AdminStudio.

Support for 64-Bit Managed-Code Custom Actions

InstallShield now has support for 64-bit managed-code custom actions. When you build a release that includes a managed-code custom action in your project, InstallShield attempts to determine the target architecture (32 bit or 64 bit) of the main .NET assembly that is associated with the custom action. InstallShield configures the release so that the appropriate version of the .NET Framework—32 bit or 64 bit—is used to run your managed code at run time.

If you want to override the new default behavior, use the Direct Editor view to add a new record with the following fields to the ISClrWrap table.

- **Action_**—Indicate the name of the managed-code custom action that you want to modify.
- **Name**—Enter the following: TargetPlatform
- **Value**—Specify the appropriate architecture. To use a 32-bit version of the .NET Framework, enter the following: x86
  To use a 64-bit version of the .NET Framework, enter the following: x64

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

This feature resolves issue IOA-000051744.

Support for 64-Bit .NET Installer Classes and COM Interop

InstallShield now has support for 64-bit .NET installer classes and COM interop. If you are using InstallShield on a 64-bit system, the .NET tab on the Options dialog box—which is displayed when you click Options on the Tools menu in InstallShield—now lets you specify different paths for the 32-bit and 64-bit locations of the Regasm.exe and InstallUtilLib.dll files that are included with the .NET Framework. InstallShield uses the paths that you specify at build time for releases that include .NET installer classes and COM interop.

If you are building from the command line with ISCmdBld.exe and you use the existing -t parameter to specify the path of the 32-bit version of the .NET Framework, ISCmdBld.exe now uses the 64-bit location of Regasm.exe and InstallUtilLib.dll when appropriate.

If you are building through MSBuild or Team Foundation Server (TFS) and you use the existing DotNetUtilPath parameter on the InstallShield task to specify the path of the 32-bit version of the .NET Framework, the build now uses the 64-bit location of Regasm.exe and InstallUtilLib.dll when appropriate.

This feature resolves issue IOA-000056407.

Support for Using the InstallScript Function DotNetCoCreateObject or Managed-Code Custom Actions with DLLs that Target .NET Framework 4

If you create a DLL in Visual Studio 2010 and the DLL uses .NET Framework 4, you can now use the InstallScript function DotNetCoCreateObject to call functions in this DLL. Previously, the installation crashed if the DLL used version 4 of the .NET Framework, but not with earlier versions. This feature is available in the following project types: InstallScript and InstallScript MSI; it is also available in Basic MSI and Merge Module projects with InstallScript custom actions.

In addition, you can now use the same sort of DLL file in a managed-code custom action. Previously, the installation crashed if it contained a managed-code custom action for a DLL that used version 4 of the .NET Framework. This feature is available in the following project types: Basic MSI, InstallScript MSI, Merge Module.

This feature resolves issue IOA-000055646.

Ability to Override Path Variables for a Release

InstallShield now lets you override the values of your project's user-defined path variables, environment variables, and registry variables for each release in your project. This functionality enables you to essentially replace certain files and folders in your project with others at build time, depending on the particular release that you are building.

For example, you might use this feature to swap out the binaries for custom actions. If you have set up separate releases for 32-bit and 64-bit target systems, you can override the path variable that is used to refer to the DLL that is selected for a custom action. Then InstallShield could include a 32-bit DLL for your 32-bit release and a 64-bit DLL for your 64-bit release. Note that overriding path variables to swap out files that your installation is installing is not recommended. This is because you should use separate components for 32-bit and 64-bit versions of a file.

To override one or more path variables in your project, use the new Path Variables Overrides setting, which is on the Build tab for a release in the Releases view. Note that if you override a path variable both in the new Path Variables Overrides setting, and with the -l parameter to IsCmdBld.exe or through MSBuild, the command line or MSBuild value takes precedence over the value that is set in the release setting.

The automation interface now includes support for this new setting. The ISWiRelease object includes a new PathVariableOverrides property that lets you override the values of your project's user-defined path variables, environment variables, and registry variables for a release.

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

Ability to Configure MIME Types for IIS Web Sites, Applications, and Virtual Directories

The Internet Information Services view has a new MIME Types setting that you can configure for a Web site, application, or virtual directory in your project. This setting lets you identify the types of content that can be served from the Web server on the target system to a browser or mail client.

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

This feature resolves issue 1-G7VZH.

Ability to Overwrite Existing IIS Application Pool or Only Create It If It Does Not Exist

The Internet Information Services view has a new Overwrite Existing Application Pool setting. This setting is displayed in the right pane when you select an application pool in the Internet Information Services view. This setting lets you specify the behavior that you want to occur if the selected application pool already exists on the target system at run time: the installation can either overwrite the application pool's settings on the target system, or leave the application pool as is. The default value for this setting is Yes, indicating the an existing application pool is overwritten at run time.

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

This feature resolves the following issues: IOA-000052593 and IOC-000066151.

New Billboard Style in InstallScript and InstallScript MSI Projects

InstallScript and InstallScript MSI projects include support for a new style of billboard that is displayed above the progress bar on the progress dialog. This style of billboard supports Adobe Flash application files (.swf), as well as image files (.bmp, .gif, .jpg, and .jpeg).

If a project contains both Flash files and image files but a target system does not have the Adobe Flash Player, the installation can detect that and display image billboards in place of the Flash billboard.

Previously, the only available billboard support in InstallScript and InstallScript MSI projects required the use of a background window. The new billboard style does not require a background window.

To add the new style of billboard to a project, use the Support Files/Billboards view to add billboard files to your project. To display this new style of billboard at run time, use the new STATUSBBRD constant with the Enable function.

Note that the new style of billboard is not available in projects that use skinned dialogs.

Support for Looping Through Image Billboards in Basic MSI Projects

The Billboards view in Basic MSI projects has a new Loop Billboards setting that lets you specify whether you want your installation to continuously loop the image billboards until the file transfer completes and the installation displays the appropriate SetupComplete dialog.

If you select Yes for this setting and the file transfer takes more time than you have allocated for the billboards, the installation restarts the display of billboards from the beginning. The loop continues, if necessary, until the file transfer ends. The default value for this setting is No, which matches the behavior in earlier versions of InstallShield.

Previously, if the file transfer took more time than what you had allocated for the billboards, the installation continued to display the last billboard until the file transfer finished; it did not loop the billboards.

This feature resolves issue IOA-000053355.

Windows Installer 5 Support for Configuring Windows Service Permissions

InstallShield now includes support for configuring permissions for Windows services; this support uses the MsiLockPermissionsEx table that was introduced with Windows Installer 5. Windows Installer 5 supports the new service permission settings; earlier versions of Windows Installer ignore them.

To configure the new permission settings for a service, use the Services node in the Advanced Settings area for a component in the Setup Design view (in installation projects) or the Components view. The Services node is where you add new services to your project. When you select a subnode under the Services node, you can configure the settings—including the new permission-related settings—of that service in the right pane.

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

Automation Interface Support for Configuring Dynamic File Links in InstallScript Projects

The ISWiDynamicFileLinking object and the ISWiDynamicFileLinkings collection of the automation interface now include support for dynamic file links in InstallScript projects. In addition, two methods and a property of the ISWiComponent object are now applicable to InstallScript projects: the AddDynamicFileLinking method lets you add a new dynamic file link to a component, the RemoveDynamicFileLinking method lets you remove a component's dynamic file links, and the ISWiDynamicFileLinkings property lets you get a component's collection of dynamic file links. The read-only DynamicFile property for the ISWiFile object is also now applicable to InstallScript projects: use this property to determine whether the file's source is linked to the component dynamically or statically.

This feature resolves issue IOC-000082900.

New FlexNet Connect 12.01 Redistributables Available

InstallShield includes support for FlexNet Connect 12.01 in Basic MSI and InstallScript MSI projects. Use the Update Notifications view in InstallShield to include one of the two FlexNet Connect 12.01 merge modules—one has the Common Software Manager, and the other does not.

Enhancements in InstallShield 2011 Original Release Version (August 2010)

Unicode Views in InstallShield

Several views and areas in InstallShield have been enhanced to display and allow you to enter characters from all languages. For example, now you can use Chinese characters in file names, paths, and registry entries if you are configuring system searches in the System Search view on an English machine. Previously, the characters were displayed in these areas of InstallShield as question marks.

The areas in InstallShield that have been enhanced to include the Unicode support are the System Search view, the tabs for a release in the Releases view, and the Multiple Instances tab for a release in the Releases view. Note that Unicode support was previously introduced in many other views in InstallShield 2010.

The improvements for the System Search view are applicable to the following projects: Basic MSI, InstallScript MSI, Merge Module, MSI Database, MSM Database, and Transform.

The improvements for the tabs for a release in the Releases view are applicable to the following projects: Basic MSI, InstallScript, InstallScript MSI, InstallScript Object, and Merge Module.

The improvements for the Multiple Instances tab in the Releases view are applicable to Basic MSI projects.

Command-Line Support for Generating a Log File While Using InstallShield MSI Diff

InstallShield MSI Diff includes a new /out command-line parameter. You can use this new parameter while running InstallShield MSI Diff from the command line to generate a log file that identifies the differences between two .msi, .msm, or .pcp files, or that shows the differences that are applied to a Windows Installer database by a transform (.mst) or patch file (.msp). You can also use this tool from the command line to generate a log file that identifies the differences between two InstallShield project files (.ism or .ise) that are saved in binary format.

This enhancement resolves the following issues: IOA-000054055, IOC-000076072.

Support for Listing Local Microsoft SQL Server Instances in the Run-Time Dialogs on 64-Bit Systems

If you add SQL support to a project through the SQL Scripts view in InstallShield and the installation is run on a 64-bit target system, the built-in SQL-related run-time dialogs now list 64-bit local SQL Server instances, as well as 32-bit local SQL Server instances, 32-bit network SQL Server instances, and 64-bit network SQL Server instances. Previously when the installation was run on a 64-bit target system, the built-in SQL-related run-time dialogs did not list any 64-bit local SQL Server instances; they listed only 32-bit local instances, 32-bit network instances, and 64-bit network instances.

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

This enhancement resolves the following issues: IOA-000054153, IOB-000056728, IOC-000053557.

InstallScript Logging and Uninstallation of 64-Bit Registry Changes

By default, the InstallScript engine now logs the changes that it makes to the 64-bit part of the registry during an InstallScript or InstallScript MSI installation on 64-bit target systems. Furthermore, the 64-bit registry changes that are logged by the InstallScript engine are now uninstalled during uninstallation.

Previously, if the InstallScript engine made changes to the 64-bit part of the registry during an installation, those changes were not logged as 64 bit specific, so they were not removed during uninstallation.

This enhancement resolves issue IOC-000046694.

Support for Microsoft SQL Server Management Studio

The SQL Scripts view now includes support for Microsoft SQL Server Management Studio. To open a SQL script file of your project in SQL Server Management Studio, right-click the SQL script file in the SQL Scripts view and then click Open Script in Microsoft SQL Server Management Studio. This new command replaces the Open Script in Microsoft Query Analyzer command on the same context menu.

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

This enhancement resolves issue IOC-000080976.

Support for Adding Project Outputs from Visual Studio Web Setup Projects

If you create a Visual Studio solution that includes a Web setup project and an InstallShield installation project and you are using InstallShield from within Visual Studio, you can now add project outputs from the Web setup project to your InstallShield project.

This enhancement resolves issue IOC-000081640.

Ability to Select a Windows Installer Property when Configuring a Text File Change

Two settings in the Text File Changes view have been enhanced to make it easier to configure search-and-replace behavior for content in text files. The Find What and Replace What settings, which are displayed in the right pane when you select a replacement node in the Text File Changes view, now let you select a Windows Installer property from a list, or type a string. The list consists of all of the properties that are available in the Property Manager view of your project. Previously, you could manually enter a string or a property in these settings, but the list of properties was not available.

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

This enhancement resolves issue IOA-000051122.

InstallScript Text Substitution Improvements for Resolving Environment Variable Paths in IISROOTFOLDER

InstallScript installations now have support for installing IIS data to IISROOTFOLDER if the IIS PathWWWRoot registry value contains an environment variable. Previously, when text substitution was performed for the IISROOTFOLDER variable in an InstallScript installation that included IIS support, the environment variable part of the path of IISROOTFOLDER was not resolved on Windows Server 2008 and later systems. If any components were configured to be installed to IISROOTFOLDER, they failed to install because the unresolved path was an invalid path.

This enhancement resolves issue IOA-000051120.

Ability to Specify the Required Execution Level for Update.exe Manifests

The Required Execution Level setting is now available on the Advanced tab for a patch configuration in the Patch Design view of Basic MSI and InstallScript MSI projects. It is also available on the Advanced tab of the Build Settings area in the General Information view of QuickPatch projects.

Use these new settings to specify the minimum execution level that is required by your installation's Update.exe file for running the upgrade on Windows Vista and later platforms. InstallShield adds a manifest that specifies the required level. By default, InstallShield uses the level that was configured in the previous setup launcher's manifest.

Previously, the Required Execution Level setting was available only for the Setup.exe setup launchers. If you created an Update.exe patch, InstallShield used the same required execution level that was configured in the previous setup launcher's manifest.

This enhancement resolves the following issues: IOA-000055255, IOC-000079301.

Ability to Create .cab Files Without Compression

A new Cab Optimization Type setting is available on the Build tab for a release that is selected in the Releases view. If Compressed or one of the custom options is selected for the Compression setting, use the Cab Optimization Type setting to specify the type of compression that InstallShield should use when building the release's .cab files. The available options are LZX compression, MSZIP compression, or no compression.

The Cab Optimization Type setting replaces the Optimize Size setting, which had support for only LZX compression and MSZIP compression; it did not let you skip compression.

The automation interface now includes support for this new setting. The ISWiRelease object includes a new CabCompressionType property that lets you specify which of the three compression options you want to use when you build a release through the automation interface.

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

This enhancement resolves issue IOA-000053420.

Expanded List of Predefined Operating System Conditions in the InstallShield Prerequisite Editor

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, now has predefined operating system conditions for Windows Server 2008 R2. Some of the Windows 7 options have been renamed to make it more clear which versions of Windows are checked, since some of the options check for the workstation version, the server version, or either version. The Prerequisite Condition dialog box also has new options that check only for Windows 7, not Windows Server 2008 R2. These changes let you quickly define more-targeted operating system conditions for any InstallShield prerequisites.

You can add InstallShield prerequisites to the following project types: Basic MSI, InstallScript, and InstallScript MSI.

This enhancement resolves issue IOA-000054066.

New Refresh Button in the Redistributables and Prerequisites Views

The Redistributables view (in Basic MSI, InstallScript MSI, and Transform projects) and the Prerequisites view (in InstallScript projects) contains a new Refresh button that you can use to refresh the list of redistributables that are displayed in the view. Previously, if you added redistributables to your machine when either view was open in InstallShield, it was necessary to close and then reopen the project in order to see an updated list in the view.

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

This enhancement resolves issue IOA-000051535.

Support for Setting Permissions for the IUSR User Account

InstallShield now has support for securing files, folders, and registry keys for the well-known user account IUSR.

If you are using the custom InstallShield handling method of configuring permissions in your project, the User list on the Permissions dialog box has a new IUSR option. To use the custom InstallShield handling method, select this option in the Locked-Down Permissions setting in the General Information view. This functionality is available in the following project types: Basic MSI, InstallScript MSI, Merge Module, and Transform.

If you are using the InstallScript function SetObjectPermissions to set permissions, you can now pass IUSR for the szUser parameter. This functionality is available in InstallScript events in the following project types: InstallScript and InstallScript MSI. This functionality is also available through InstallScript custom actions in Basic MSI, InstallScript MSI, and Merge Module projects.

This enhancement resolves issue IOC-000080797.

New IS_PERMISSIONS_OPTION_ALLOW_ACCESS Constant for the SetObjectPermissions Function

A new IS_PERMISSIONS_OPTION_ALLOW_ACCESS constant is available for use with the InstallScript function SetObjectPermissions. You can pass this constant instead of the number 0 in the nOptions parameter of SetObjectPermissions to indicate that the permissions that are being set should allow access to the specified file, folder, or registry key.

This functionality is available in InstallScript events in the following project types: InstallScript and InstallScript MSI. This functionality is also available through InstallScript custom actions in Basic MSI, InstallScript MSI, and Merge Module projects.

This enhancement resolves issue IOC-000081824.

New Constants for the DialogSetInfo Function

Two new constants are available for the nParameter parameter of the InstallScript function DialogSetInfo when you are using DLG_INFO_ALTIMAGE for the nInfoType parameter:

- **DLG_INFO_ALTIMAGE_VERIFY_BMP**—This constant specifies that the bitmap that is indicated by szInfoString should be used in subsequent dialogs. Before this bitmap is used, the installation checks for the existence of the bitmap. If the bitmap does not exist, the function returns an error that indicates that the bitmap was not found.

- **DLG_INFO_ALTIMAGE_REVERT_IMAGE**—This constant specifies that dialogs should display the default bitmap. This constant is an alternative equivalent to the value of -1. The installation does not check for the existence of the bitmap when you use this constant.

If you pass TRUE in the nParameter parameter, the installation does not check for the existence of the bitmap; if the bitmap does not exist in this scenario, the function does not return an error.

This functionality is available in InstallScript events in the following project types: InstallScript and InstallScript MSI.

This enhancement resolves issue IOA-000054733.

InstallScript Language Enhancements for .NET Framework 4.0 Support

A new FOLDER_DOTNET_40 InstallScript variable is available. This variable stores the path of the .NET Framework 4.0 files. Two new constants are available for use with the function Is:

- REGDB_KEYPATH_DOTNET_40_CLIENT
- REGDB_KEYPATH_DOTNET_40_FULL

Ability to See Details about File and Registry Sets for a Project's DIM References

When you select a DIM reference in the DIM References view, two new tabs are displayed:

- **File Sets**—This tab shows read-only details about all of the file sets that are referenced by the selected DIM. This includes static files, dynamically linked files, and empty directories.
- **Registry Sets**—This tab shows read-only details about the registry sets that are referenced by the selected DIM.

These enhancements are available in the following project types: Basic MSI and Merge Module.

Path Variable Support for DIM References

When you add DIM references to a project, InstallShield now lets you use path variables for the paths of .dim files instead of absolute paths. This support makes it easier to build releases on different build and development machines that use different source file locations. The General tab in the DIM References view lists the path, including any applicable path variables, for the selected .dim file.

This support is available if InstallShield is configured to use path variables on your system. (That is, the Path Variables tab on the Options dialog box must be configured to allow path variable support.) If you configure InstallShield to use absolute paths, the General tab in the DIM References view shows the absolute path to .dim files.

This enhancement is available in the following project types: Basic MSI and Merge Module.

This enhancement resolves issue IOA-000053026.

Path Variable Support for Each Patch Configuration's Output Location and Cache

The Common tab for a patch configuration that is selected in the Patch Design view contains two settings that let you browse to the appropriate folder: Patch Output Location and Patch Creation Cache. InstallShield now lets you use path variables for the folders instead of absolute paths. This support makes it easier to build releases on different build and development machines that use different directory structures.

This support is available if InstallShield is configured to use path variables on your system. (That is, the Path Variables tab on the Options dialog box must be configured to allow path variable support.) If you configure InstallShield to use absolute paths, the Patch Design view shows the absolute path to folders.

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

This enhancement resolves issue IOA-000054041.

InstallScript Functions ServiceExistsService and ServiceGetServiceState No Longer Require Elevated Privileges

The InstallScript functions ServiceExistsService and ServiceGetServiceState no longer require elevated privileges. Therefore, installations can now call these functions when end users have limited privileges, such as in the Install UI sequence of a Basic MSI installation.

This enhancement resolves issue IOC-000059857.

Important Information

Evaluating InstallShield

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 or connecting it to a license server. When you use InstallShield before activating it or connecting it to a license server, it operates in evaluation mode, and some of its functionality is not available. For details, see KB article Q200900. Note that the evaluation limitations are removed when you activate InstallShield or when you connect it to a license server and check out a license for it.

Obtaining the InstallShield Standalone Build

The installation for the Standalone Build consists of a single, compressed executable file that you can obtain through either of the following methods:

- If you have the InstallShield DVD, the file is on the DVD and you can find it using the DVD Browser.
- If you downloaded InstallShield, the Standalone Build installation file is available for download as documented in the Standalone Build licensing instructions.

For detailed information on license configuration for the Standalone Build, see the Standalone Build licensing instructions.

Installing More than One Edition of InstallShield

Only one edition of InstallShield 2011—Premier, Professional, or Express—can be installed on a system at a time.

Installing More than One Version of InstallShield

InstallShield 2011 can coexist on the same machine with other versions of InstallShield.

The InstallShield 2011 Standalone Build can coexist on the same machine with other versions of the Standalone Build. In most cases, the Standalone Build is not installed on the same machine where InstallShield is installed. If you do install both on the same machine and you want to use the automation interface, review the "Installing the Standalone Build and InstallShield on the Same Machine" help topic in the InstallShield Help Library to learn about special registration and uninstallation considerations.

Integrating InstallShield with Visual Studio

Microsoft Visual Studio can be integrated with only one version of InstallShield 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.

End of Integration Support for Visual Studio 2003 and Earlier

If you want to create, edit, and build your InstallShield projects directly within Visual Studio, you must use Visual Studio 2005 or later. InstallShield can no longer be integrated with Visual Studio 2003 or earlier.

MSBuild Integration

If you use MSBuild to build Visual Studio solutions with InstallShield projects, MSBuild requires .NET Framework 3.5 or later.

Deprecation of InstallScript Objects

InstallScript objects have been deprecated in favor of InstallShield prerequisites. In a future release, InstallShield will no longer be able to create or consume InstallScript objects, and no predefined InstallScript objects will be provided. Furthermore, the Merge Module Holder Object will not be available. The recommended alternative for InstallScript objects is InstallShield prerequisites. You can use the InstallShield Prerequisite Editor to begin creating your own InstallShield prerequisites so that you are ready to use them once the InstallScript object technology becomes obsolete. You can share these InstallShield prerequisites among InstallScript, InstallScript MSI, and Basic MSI projects.

Project Upgrade Alerts

The following information describes possible upgrade issues that may occur when you upgrade projects that were created with InstallShield 2010 and earlier to InstallShield 2011. It also alerts you to possible changes in behavior that you may notice between new InstallShield 2011 projects and projects that are upgraded from InstallShield 2010 or earlier to InstallShield 2011. For updates to this information, see Knowledge Base article Q208911.

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

If you use InstallShield 2011 to open a project that was created with an earlier version, InstallShield 2011 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 .770 before converting it. Delete the .770 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 2011 projects in earlier versions of InstallShield.

You can upgrade projects that were created with the following versions of InstallShield to InstallShield 2011: InstallShield 2010 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 2011.

InstallShield Builds Only Unicode Versions of Setup.exe and Update.exe; Ability to Create ANSI Versions Is No Longer Available

Now all Setup.exe and Update.exe files that are built in all project types in InstallShield are Unicode. This applies to all new Basic MSI, InstallScript, InstallScript MSI, and QuickPatch projects that you create in InstallShield 2011. It also applies to all projects that you have upgraded from earlier versions of InstallShield to InstallShield 2011. Therefore, the settings that previously enabled you to specify whether you wanted to build a Unicode version or an ANSI version of the setup launcher have been removed:

- The Setup Launcher Type setting on the Setup.exe tab for a release in the Releases view has been removed from Basic MSI projects.
- The Update Launcher Type setting was removed from Basic MSI, InstallScript MSI, and QuickPatch projects. In Basic MSI and InstallScript MSI projects, this setting was on the Advanced tab for a patch configuration in the

Patch Design view. In QuickPatch projects, this setting was on the Advanced tab in the Build Settings area of the General Information view.

Changes to Win32 API Definitions

Now that the InstallScript engine has been updated to support Unicode, the Win32 API functions that are prototyped in the InstallScript header file ISRTWindows.h have been updated. Where applicable, wide (W) versions of API prototypes have been added in addition to existing ANSI (A) definitions. For some prototypes, no A or W version is specified; in these cases, the engine now attempts to use the W version. Previously, the A version was used.

If you upgrade an InstallShield 2010 or earlier project to InstallShield 2011, it is possible that these new prototypes could conflict with user-defined prototypes of the same APIs. It is recommended that the InstallScript-provided prototypes be used if possible. However, if you want to use your own prototypes for these Windows APIs instead of the ones that are now prototyped for InstallScript, add ISINCLUDE_NO_WINAPI_H 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_WINAPI_H. Otherwise, you may encounter compile errors.

Ensuring that Your InstallScript Code Supports Unicode

If you use any user-defined Win32 APIs or other external DLL prototypes in your InstallScript code, and the APIs have versions that support Unicode string input, update the prototypes to use BYVAL/BYREF WSTRING or WPOINTER. In addition, review the API functions in your code that accept structures as input to ensure that those that contain string or string pointer members are declared as Unicode as appropriate, if the API requires it.

Update your script to call W versions of Win32 APIs if it is currently calling A versions.

Differential Release Support in InstallScript Projects

A differential release that is created in an InstallShield 2011 or later InstallScript project can update a product only if its earlier InstallScript installation was also created in InstallShield 2011 or later. If you want to use InstallShield 2011 or later to create an update for a product whose earlier InstallScript installation was created with InstallShield 2010 or earlier, you should create a full release, instead of a differential release.

This requirement is necessary because of the Unicode support that is new in the InstallShield 2011 InstallScript engine. After a differential release is used to update an earlier version of a product on a target system, the earlier version of the installation (along with the earlier engine run time) is used to run any maintenance operations. Because the InstallShield 2010 and earlier versions of the InstallScript engine cannot read the new Unicode storage format, the installation fails.

Uninstalling 64-Bit Registry Entries that Were Installed by the InstallScript Engine

By default, the InstallScript engine now logs the changes that it makes to the 64-bit part of the registry during an InstallScript or InstallScript MSI installation on 64-bit target systems. Furthermore, the 64-bit registry changes that are logged by the InstallScript engine are now uninstalled during uninstallation.

Note that if you created an InstallShield 2010 or earlier installation that wrote 64-bit registry data and you create an upgrade for your product in InstallShield 2011, the existing 64-bit registry entries that were logged by the base installation are not removed when the product is uninstalled from a target system. The only way to work around this limitation is to manually remove the registry data during uninstallation.

Script Editor Changes

Pressing CTRL+SPACEBAR from within the script editor pane in the InstallScript view no longer displays a list of built-in InstallScript functions that are available for auto completion, as it did in InstallShield 2010 and earlier. Now

if you want to see a pop-up list of built-in InstallScript functions for auto completion, you simply need to start typing the first letter or letters of that function. The pop-up list also contains other InstallScript keywords.

The context menu that is displayed when you right-click in a script editor pane is now different than it was in InstallShield 2010 and earlier:

- The context menu no longer contains a Show Whitespace command. To show or hide whitespace characters and symbols in the script editors, use the Script Editor Properties dialog box. This dialog box also lets you modify other settings in the script editors, such as font, syntax colors, and line numbering. To access this dialog box, right-click in a script editor and then click Properties.
- The context menu no longer contains a Make Uppercase command or a Make Lowercase command. To make text in the script editor uppercase, select it and then press CTRL+SHIFT+U. To make it lowercase, select it and then press CTRL+U.
- The context menu no longer contains a Find command or a Replace command. To perform a search in a script, press CTRL+F, and then specify the text that you want to find. To search for a string and replace it with something else, press CTRL+H and then specify the appropriate information. As an alternative, you can use the Find and Replace commands on the Edit menu.

The views that contain the revised script editor are the InstallScript view, the SQL Scripts view, and the Custom Actions and Sequences view (when you are viewing a VBScript or JScript file in this view).

InstallScript Debugger Changes

If you copy the InstallScript Debugger (ISDbg.exe) from your installation development machine to a debug machine so that you can debug InstallScript code, you must also now copy the file called SciLexer.dll to the same folder on the debug machine. You can find SciLexer.dll on your installation development machine in the same folder as the ISDbg.exe file (*InstallShield Program Files Folder*\System).

DPI Changes and Their Effect on InstallScript Dialogs

If you have an InstallShield 2010 or earlier InstallScript or InstallScript MSI project that contains one or more edited InstallScript dialogs and you want to upgrade the project to InstallShield 2011 or later, ensure that your machine uses the same DPI value that was selected when the InstallScript dialogs were edited. Otherwise, the dialogs may be sized incorrectly at run time.

Note that once you have upgraded the project, you can change the DPI value on your machine at any time as needed; if you do, the dialogs are sized correctly at run time. Also note that if you upgrade an InstallShield 2011 or later project that contains one or more edited InstallScript dialogs to a future version of InstallShield, you do not need to use the same DPI value during the upgrade.

Changing Design-Time and Build-Time Locations of Existing InstallShield Prerequisites in Existing Projects

InstallShield now lets you specify the folders where InstallShield should search for InstallShield prerequisite files (.prq files), their associated data files, and their dependencies. Previously, InstallShield searched for .prq files in the following location only: *InstallShield Program Files Folder*\SetupPrerequisites.

If you move any InstallShield prerequisites from the *InstallShield Program Files Folder*\SetupPrerequisites folder to a new custom location that you have defined on the Prerequisites tab of the Options dialog box (or any of the other places where search paths can be defined now), you may need to perform the following steps in InstallShield 2010 or earlier projects when you upgrade them to InstallShield 2011:

1. In the Redistributables view or the Prerequisites view, clear the check box for each InstallShield prerequisite that is included in your project but is located in a custom location. Also clear the check box for each

InstallShield prerequisite whose data files or dependencies were moved from the default location to a custom location.

2. Click the new Refresh button.
3. Select the check box for each InstallShield prerequisite that you removed from your project in step 1.

InstallShield removes the path of the prerequisite from the ISSetupPrerequisites table of your project. The full path was stored in this table in InstallShield 2010 and earlier projects. Note that if you only clear a prerequisite's check box and then reselect it without clicking the Refresh button, InstallShield continues to use the full path, rather than just the file name, in the ISSetupPrerequisites table.

If you upgrade an InstallShield 2010 or earlier project to InstallShield 2011, change the location of an InstallShield prerequisite, and then add that prerequisite to your project, you do not need to perform the refresh procedure. Also, if you create a new project in InstallShield 2011, you do not need to perform the refresh procedure. In both cases, InstallShield does not include the path in the ISSetupPrerequisites table of your project, which enables you to use the custom search path, instead of the default path.

Preventing an InstallScript MSI Installation from Overwriting a Future Major Version of the Same Product

The Upgrades view in new InstallScript MSI projects now contains a major upgrade item called ISPreventDowngrade. This item prevents end users from being able to install the current version of your product over a future major version of the same product. If you upgrade an InstallScript MSI project from InstallShield 2010 or earlier to InstallShield 2011, the ISPreventDowngrade item is not added automatically. You can manually add an ISPreventDowngrade item if appropriate. To learn how, see the "Preventing the Current Installation from Overwriting a Future Major Version of the Same Product" help topic in the InstallShield Help Library.

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 2011, you can still do so. For instructions, see Knowledge Base article Q200884.

Resolved Issues in InstallShield 2011 Original Release Version (August 2010)

1-12UC1P (InstallScript)

If the language dialog is displayed at run time, pressing the up and down arrow keys to move through the language list now behaves as expected.

1-KBTG5

The Serial Number setting on the CustomerInformation dialog that is displayed during the InstallShield installation is now a single field; this enables you to copy your entire serial number to your Clipboard and then paste it into this field. Previously, this field was a multipart field. The copy-paste operation required that you copied and pasted each part of the serial number separately.

IOA-000030302 (QuickPatch)

Building a cumulative QuickPatch project that updates a read-only file no longer generates build error -1007

IOA-000033311 (Basic MSI, InstallScript MSI)

If you use the InstallScript variable DISK1SETUPEXENAME in an immediate-mode custom action and an end user runs your installation by launching the Setup.exe setup launcher, the DISK1SETUPEXENAME variable now contains the file name and file name extension of the setup launcher. Previously, the variable was blank.

IOA-000044450

The InstallScript function PathDelete now handles decimal points in paths correctly. Previously in some cases, using this function to delete a specific directory in the path buffer may have erroneously deleted a path if it included a period.

IOA-000047710

If you modify the Text setting of an edit field control on a dialog in your project and then you delete the corresponding string identifier and value from your project, InstallShield generates a unique string identifier such as NEW_STRING## (where ## is a unique number) and associates that string identifier with the control. You can change the string identifier and its value as needed. Previously in this case, InstallShield assigned the control a value of ***IS_STRING_NOT_DEFINED***, and the only way to remove it from the project was to use the Direct Editor view to delete it from the Control table.

IOA-000048528 (MSI Database)

InstallShield no longer crashes when you save an MSI Database project after you have added some files. Previously in some cases, InstallShield crashed.

IOA-000048747 (InstallScript)

An InstallScript installation that has a major version number greater than 127 can now be used to upgrade a product that has a major version less than or equal to 127.

IOA-000049470 (InstallScript)

If No is selected for the Uninstall setting of a component that contains a font, that font is now permanently left on the target system. Previously, uninstalling the product or the feature that contained the component also uninstalled the font.

IOA-000049854 (InstallScript)

If you use the Japanese version of InstallShield to build an InstallScript One-Click Install installation, the default Web page (Setup.htm) that InstallShield generates is now displayed correctly on Japanese systems. Previously, the page displayed garbled text.

IOA-000050400 (QuickPatch)

InstallShield no longer crashes if you try to build a QuickPatch package in which Yes is selected for the "Create new 'UpgradedImage' folder" setting and if patching and upgrading validation is turned on.

IOA-000050926 (Standalone Build)

The Standalone Build no longer generates fatal error -1013 at build time when you are trying to use the same Standalone Build instance to build releases from two separate projects.

IOA-000051118, IOB-000059480 (InstallScript)

The built-in InstallScript objects now include the necessary language resources to function correctly at run time in multilanguage InstallScript installations that include languages other than English.

IOA-000051393 (InstallScript, InstallScript MSI)

The Korean translation of two string entries that are used on the SdFeatureTree dialog and one string entry that is used on the exit setup message box have been corrected.

IOA-000051828 (Basic MSI, Merge Module)

When you convert a Visual Studio project (.vdproj) into an InstallShield project (.ism), the language information is now converted correctly. Previously in some cases, the language information was not converted correctly.

IOA-000051894 (Basic MSI, InstallScript MSI)

When you import a .reg file into your project, the registry data now have the asterisk flag (*), which indicates that it will be installed on the target system if it is absent but uninstalled if it is present. This applies to imported registry keys that have no keys or values under them. Previously in some cases, some of the registry data had that flag, but some of it may have had the plus flag (+), which indicated that it would be installed on the target system.

IOA-000051938

In the "Data Types and Predefined Structures" help topic, the description of *VARIANT* now explains that the VARIANT data type cannot be defined within a data structure. Previously, this information was not documented.

IOA-000052023 (Merge Module)

If you add a DIM reference to a Merge Module project and the DIM contains dependencies, InstallShield no longer crashes.

IOA-000052103 (InstallScript)

If you create an InstallScript installation that includes a Setup.gif splash screen, the splash screen is now displayed at run time. Previously, the splash screen was not displayed for .gif files; only .bmp files could be used.

IOA-000052186 (Basic MSI, InstallScript MSI)

If VarSave or VarRestore (the InstallScript functions) are called in an InstallScript custom action, the value of INSTALLDIR is changed only if the SRCINSTALLDIR, VAR_SRCTARGETDIR, or SRCTARGETDIR constant is passed for the nType parameter. Previously in some cases, the value of INSTALLDIR was changed, regardless of whether one of these constants were passed.

IOA-000052187 (Basic MSI, Merge Module)

If you add a reference to a DIM file that contains a .NET assembly whose Contains Installer Class setting is set to True, the custom actions that call the installer class functions are now added to the Basic MSI or Merge Module release at build time. Previously, the custom actions were not added.

IOA-000052226 (InstallScript)

The Complete radio button is now selected by default in the SdSetupType2 dialog of a Romanian installation. Previously, no radio button on this dialog was selected by default.

IOA-000052229 (Basic MSI, InstallScript MSI, Smart Device)

If you include the Microsoft SQLCE 3.1 redistributable in your mobile device installation, the Application Manager no longer displays an error at run time stating that it cannot install the application on the mobile device.

IOA-000052233 (Merge Module)

InstallShield truncates the name of a DIM reference in a Merge Module project if the name contains more than 65 characters. (An additional 7 characters are reserved for the unique Windows Installer key.) Previously, the name was truncated if it contained more than 28 characters.

IOA-000052246 (Basic MSI, InstallScript MSI, Merge Module)

If you are using the Reg-Free COM Wizard and you specify a manifest that is not named properly for the corresponding COM server, InstallShield no longer crashes.

IOA-000052360 (Basic MSI, InstallScript, InstallScript MSI)

If you enter a value in the ASP Script Timeout (seconds) setting in the Internet Information Services view, the installation now properly sets the timeout value at run time.

IOA-000052365 (Merge Module)

If you add a DIM reference to a Merge Module project and the DIM contains a dynamic file set, the dynamically linked files are now included in the release.

IOA-000052410 (Basic MSI with InstallScript custom actions, InstallScript, InstallScript MSI)

Installations that are built from an InstallScript project no longer fail during setup initialization with error -5009. In addition, installations that are built from an InstallScript MSI project or from a Basic MSI project containing InstallScript custom actions no longer fail with error 1628 on Windows 2000 systems. Previously in these cases, these run-time errors occurred when the installations were run on Windows 2000 systems.

IOA-000052500 (InstallScript, InstallScript MSI)

If certain types of dialog controls (such as the selection tree on the SdFeatureTree dialog) are disabled on an InstallScript dialog, the installation no longer encounters run-time error 0x80040707 on a Windows 2000 system.

IOA-000052533 (InstallScript, InstallScript MSI)

If you are building a release that includes a language whose name contains characters that are not available in the default code page on your machine, and you have edited any of the dialogs in your project, build error -6118 no longer occurs.

IOA-000052639 (Basic MSI)

When a second instance of a multilingual, multiple-instance, compressed installation is being run, the Command Prompt window is no longer displayed. Previously, the Command Prompt window opened, showing Msiexec.exe help, if the .msi package was compressed in a location that contained spaces.

IOA-000052661 (Basic MSI)

When a log file is being created at run time and the installation exits before the CostFinalize action, you can now use the ShowMsiLog entry point in the SetAllUsers.dll file to show the log file. This can be particularly useful if you are trying to use Windows Installer logging to debug complicated launch conditions or other logic that can fail before costing is completed, since otherwise the log file is not shown.

By default, the ShowMsiLog custom action is an executable-file custom action that launches Notepad to open the log file. If you use this default implementation and the installation stops in the UI sequence before the CostFinalize action, run-time error -2732 ("Directory Manager not initialized") occurs. To avoid this run-time error, consider using the Custom Action Wizard to edit the ShowMsiLog custom action and configure the following settings:

- **Type**—Call a function in a Windows Installer dynamic-link library
- **Location**—Stored in the Binary table
- **Source**—<ISProductFolder>\redist\language independent\i386\SetAllUsers.dll
- **Target**—ShowMsiLog

Note, however, that with this alternative implementation, the log file may be opened in a Notepad window behind other windows.

IOA-000052853 (InstallScript MSI)

If you use the Text File Changes view to configure a text replacement for a long text string, the resulting installation no longer crashes.

IOA-000052856

The Customize Validation Settings dialog box now shows appropriate descriptions of ICE101 through ICE105. (To access this dialog box: On the Tools menu, click Options. Then, on the Validation tab, click the Customize button.) Previously, this dialog box listed "User defined ICE validation rule."

IOA-000052895

In the InstallShield Help Library, the SRCDIR index entry now points to the "SRCDIR" help topic. Previously, it pointed to the wrong help topic.

IOA-000052897 (InstallScript)

If you create an InstallScript One-Click Install installation that is digitally signed with an official digital certificate (not a test certificate), end users no longer encounter a security warning.

IOA-000052924 (InstallScript)

When an InstallScript installation that includes InstallShield prerequisites is run silently, the InstallShield prerequisites are now installed if they need to be installed, and the prerequisite user interface is not displayed. Previously, the InstallShield prerequisites were not installed.

IOA-000052944 (Basic MSI, InstallScript MSI)

If the product name contains an ampersand, the title bar of each run-time dialog that is displayed by Setup.exe no longer contains an extra ampersand. This applies to dialogs such as the setup prerequisite dialog and the setup initialization dialog.

IOA-000052988

In the "Setup.exe and Update.exe Command-Line Parameters" help topic of the InstallShield Help Library, the description of the /hide_usd command-line parameter is now correct. It now states that when this command-line option is used and an update-enabled installation detects multiple previous instances, the installation creates a new instance.

IOA-000053015

Using MSBuild to build a solution that includes an InstallShield project no longer causes an MSBuild error about the InstallShield.targets file not being found. Previously this error occurred in some cases when a multiple versions of InstallShield were installed on the same system.

IOA-000053060

The "Creating a Shortcut to Check for Updates" help topic in the InstallShield Help Library has been updated. Previously, this help topic listed outdated values for the Target and Arguments settings.

IOA-000053125 (Basic MSI)

If you set the ISSCRIPTDEBUGPATH property value through the command line while launching a Basic MSI installation in debug mode, the InstallScript Debugger now opens and launches the immediate-execution InstallScript custom actions in the installation. Previously, the InstallScript Debugger was not launched. Note that

the InstallScript Debugger requires elevated privileges; thus, it is recommended that you launch the Command Prompt window with elevated privileges and then launch the installation in debug mode from that elevated Command Prompt window.

If you need to debug deferred, commit, or rollback InstallScript custom actions, any path that is specified for the ISSCRIPTDEBUGPATH property value through the command line is not accessible to these custom actions. Therefore, you must add the path of the Setup.dbg file to the registry of the system on which you are debugging. To do so, create the following registry key:
HKCU\Software\InstallShield\ISEngine*Major.Minor*
Major is the major version number of InstallShield and *Minor* is the minor version number of InstallShield.

For example, for InstallShield 2011, create the following registry key if it is not already present:
HKCU\Software\InstallShield\ISEngine17.0
In addition, add a string value called **DebugPath** to that registry key, and set its value data as the path to the Setup.dbg file.

IOA-000053147

When you use the InstallShield Prerequisite Editor to specify a dependency for an InstallShield prerequisite, the InstallShield Prerequisite Editor no longer truncates the file name of the dependency. Previously, the InstallShield Prerequisite Editor truncated the name of the .prq dependency if it contained more than 62 characters; this occurred if you used the Open dialog box to browse to and select the .prq dependency. If you added the prerequisite that contained the dependency to an installation project, the truncated file name caused a build error.

IOA-000053160 (Basic MSI, InstallScript MSI)

Now if you clear the "Delete device media from desktop during desktop setup uninstall" check box on the Desktop Settings panel in the Windows Mobile Wizard, the Windows Mobile installation is not uninstalled from the desktop when the desktop product is uninstalled. Previously, the Windows Mobile installation was uninstalled, regardless of the state of that check box on the Desktop Settings panel.

IOA-000053172 (Basic MSI, InstallScript MSI)

You can now use the BuildPCPFile method of the ISWiProject object to build a patch package (.msp) based on the settings in a patch creation properties file (.pcp). Previously, the patch build failed with error -1072803327 (ERROR_PCW_INTERNAL_ERROR).

IOA-000053189, IOA-000053383 (Basic MSI, InstallScript MSI)

The default Setup.exe strings are once again stored in the .msi package if a Setup.exe file is built for a multilanguage release. This resolves various language-related run-time issues. For example, if you build an InstallScript MSI release that includes a Setup.exe file, if the default language is not English, and if the end user launches the .msi package directly without a transform, the error message stating that the installation must be run from Setup.exe is now displayed. Previously for installations that were built in InstallShield 2010, a string such as ##StringID## was displayed instead of the error message. Another example involves minor upgrades for Basic MSI projects that contain multiple languages. The minor upgrade no longer installs new shortcuts with names such as ##ID_STRING_1##.

IOA-000053205

InstallScript functions no longer disappear from the Functions list in the center pane of the InstallScript view when the project and related files are stored on a network and InstallShield is installed on a non-English system.

IOA-000053272 (Basic MSI, InstallScript MSI, Merge Module)

If you include a dynamically linked folder that includes subfolders, and the names of two subfolders contain the same number of multibyte characters that are not supported by the system code page, the subfolders and any files within the subfolders are now installed as expected on the target system at run time. Previously, InstallShield did not properly create unique components for the two subfolders, so only one of the subfolders was created at run time, and the all of the files in both of those subfolders were installed to that subfolder.

IOA-000053336 (Standalone Build)

The InstallShield.Interop.Msi.dll and InstallShield.Interop.Stg.dll files are now installed when you run the Standalone Build installation. Previously, these files were not installed; therefore, building a release that referenced this assembly resulted in a build error.

IOA-000053374 (Basic MSI, InstallScript MSI)

In the Registry view, if you set the value data of a DWORD value to a property such as [MYPROPERTY], and then you double-click the value to edit the value data, the property is retained now. Previously, the value data was changed to the number 0 when you double-clicked the value.

IOA-000053402 (Basic MSI, InstallScript MSI)

When the SetupError dialog is displayed at run time, the error message in this dialog now contains a space after the period but before the error string. For example, the SetupError dialog may now display a string such as "Error 1327. Invalid drive[1]" instead of "Error 1327.Invalid drive[1]".

IOA-000053573

In the InstallShield Help Library, the "ConfigFileLoad" help topic now contains accurate information. It now states that the InstallScript function ConfigFileLoad cannot be used to create a new configuration file; previously, this help topic incorrectly stated that this function could be used to create a new configuration file. This help topic also now explains that to create a new configuration file, you can use CreateFile and CloseFile to create an empty file, and then use ConfigFileLoad and other functions to load and modify the file as needed.

IOA-000053590

In the InstallShield Help Library, the "AddFolderIcon" help topic now contains a modified description of the szWorkingDir parameter. It now states that AddFolderIcon writes the szWorkingDir directory in the Start In box on the Shortcut tab of the shortcut's Properties dialog box. If you pass a null string ("") in this parameter, the function leaves this Start In box blank, and the path in the Target box is used. This help topic previously contained misleading information for this parameter.

IOA-000053731 (InstallScript MSI)

The Upgrades view in new InstallScript MSI projects now contains a major upgrade item called ISPreventDowngrade. This item prevents end users from being able to install the current version of your product over a future major version of the same product.

IOA-000053800 (Transform)

If you use the Script tab in the Custom Actions and Sequences view to edit the code of a VBScript or JScript custom action that is stored in the Binary table of your transform project, the code is now displayed correctly the next time that you open your project and view the script on the Script tab. Previously, the code was not displayed correctly.

IOA-000053840 (Virtualization)

If InstallShield is unable to acquire an icon resource for a shortcut, it no longer creates an invalid .ico reference when it builds an App-V package. Previously, the App-V manifest and .osd file contained an invalid reference to

an .ico file. As a result, when the package was published through the Microsoft System Center Configuration Manager console, a "file not found" error occurred, and it was logged in the SMSAdminUI.log file.

IOA-000053950, IOB-000059050 (Basic MSI, InstallScript MSI)

If you manually enter a physical path in the Content Source Path (Local or UNC) setting for a Web site, application, or virtual directory in the Internet Information Services view, InstallShield now creates a new corresponding Directory table entry, as occurred with InstallShield 2009. Previously, with InstallShield 2010, the Directory table entry was not created; this resulted in a run-time error.

IOA-000053978 (Smart Device)

Visual Studio no longer crashes if you use InstallShield from within a Visual Studio solution that contains a Microsoft Smart Device CAB project and an InstallShield Smart Device project.

IOA-000054021 (Basic MSI, InstallScript MSI, Merge Module)

The CustomActionData property is now included as one of the available input properties for managed-code custom actions; this enables you to select this property without having to manually enter it.

IOA-000054040 (Basic MSI, InstallScript, InstallScript MSI, Merge Module)

Passing one or more of the following constants in the nPermissions parameter for a call to the InstallScript function SetObjectPermissions no longer causes compiler error C8025: STANDARD_RIGHTS_ALL, STANDARD_RIGHTS_READ, STANDARD_RIGHTS_WRITE, or STANDARD_RIGHTS_EXECUTE. Previously, the compiler error occurred because these constants were not defined in ISRTWindows.h.

IOA-000054270

The documentation for the MSI Package File Name setting has been corrected; the documentation no longer says that the value that is entered in this setting is used to name .cab files.

IOA-000054310 (Basic MSI, InstallScript, InstallScript MSI)

If you create a project that uses the Internet Information Services view to configure handler mappings for an IIS 7 application, only new handler mappings are configured on the target system at run time. Previously, all new handler mappings were configured, but existing handler mappings were also configured.

IOA-000054312 (InstallScript)

If you create an installation from an InstallScript project that uses the SQL support in the SQL Scripts view, the installation now skips the SQL server connections that target a server with an empty name. This matches the behavior that was present with InstallShield 2009 and earlier projects. With InstallShield 2010, installations attempted to connect to such a server and run the associated SQL scripts.

IOA-000054375 (Basic MSI, InstallScript, InstallScript MSI, Merge Module, Transform)

Display issues no longer occur in the Shortcuts view if you add shortcuts to your project, and the shortcuts use icons that are in executable files that contain a lot of icons.

IOA-000054479 (Basic MSI, InstallScript, InstallScript MSI, Merge Module)

Build error -6213 no longer occurs under certain conditions. Previously, this build error may have occurred if the default value of Dependencies and Properties was selected for the .NET Scan at Build setting of a component and if the component's .NET assembly had a large number of dependencies.

IOA-000054485 (Basic MSI, InstallScript, InstallScript MSI)

If you launch an installation from within its InstallShield project but the latest build does not contain the latest changes that have been made in the InstallScript view, InstallShield now recompiles the script before running the installation. Previously, if you tried to launch the installation in this case, nothing happened.

IOA-000054509 (Basic MSI, InstallScript MSI, Merge Module, Transform)

It is now possible to add to a component a service that has one or more spaces in its name. Previously, InstallShield displayed an error if you tried to use spaces in a service name.

IOA-000054648 (InstallScript, InstallScript MSI)

Run-time strings for dialog controls are no longer truncated under certain conditions. Previously in some cases, if you edited a dialog by doing something such as modifying the width of a dialog control, the string for that dialog control may have been truncated erroneously.

IOA-000054693 (InstallScript)

If you override the timestamp server that is configured in the Settings.xml file that is installed with InstallShield, InstallShield now uses the value that you have entered when digitally signing InstallScript releases at build time. The default XML contains the following:

<DigitalSignature Timestamp="http://timestamp.verisign.com/scripts/timstamp.dll"/>

To use a different timestamp server, change the value of the Timestamp attribute as needed. To disable timestamping, use an empty ("") value for the Timestamp attribute. Note that this new setting is a machine-wide setting.

Previously, the timestamp override was available in Basic MSI and InstallScript MSI projects, but not in InstallScript projects.

IOA-000054785 (Basic MSI, InstallScript, InstallScript MSI)

The New Language Wizard now lets you add support for languages that have a high language ID number. Previously, if you tried to use the New Language Wizard to add support for certain languages to a project, the wizard displayed an error stating that the language could not be added to the project.

In addition, if you try to use this wizard to enable support for such languages in new projects, and then you try to add such a language to a new project, InstallShield now lets you add that language. Previously in this scenario, InstallShield crashed.

IOA-000054845 (Basic MSI)

When you select an edit field control in a dialog that is displayed in the Dialogs view, the Property Is Integer setting is displayed in the right pane with the other settings for this control. Previously, this setting was not listed for the edit field control.

IOA-000054872 (Basic MSI, InstallScript, InstallScript MSI, InstallScript Object, Merge Module)

If you enter a password for a digital certificate on the Signing tab for a selected release in the Releases view, the password that you enter is now retained in this setting. Previously, if you entered a password for one release, selected a different release, and then returned to the initial release, the password setting was blank unless you had also saved the InstallShield project (.ism).

IOA-000055016 (Basic MSI)

You can now use the /instance command-line parameter for Setup.exe while setting the Windows Installer property TRANSFORMS at the command line for a multiple-instance installation (for example: Setup.exe /instance=1

/v"TRANFORMS=myTransform.mst"). Previously, only the /instance parameter was used. The TRANSFORMS property was ignored.

IOA-000055352 (Basic MSI, InstallScript MSI)

If you use release flags to exclude a feature from a release at build time and the feature includes a component that contains dynamically linked files, InstallShield no longer scans the dynamic file links in that component at build time. Previously, InstallShield scanned the dynamic file links at build time, which caused build errors if the dynamically linked folders were not present.

IOA-000055353

The "Installing the Standalone Build on a Build Machine" help topic in the InstallShield Help Library now lists the correct paths for the ISAppServices.tlb and ISUpgrade.tlb files. They are both installed to *Program Files Folder*\Common Files\InstallShield\Shared. Previously, the documentation had the wrong paths.

IOA-000055379 (Basic MSI, InstallScript, InstallScript MSI)

The .NET Framework 4 is now listed in the ASP.NET Version setting and the ASP.NET Platform setting for a selected Web site, application, or virtual directory in the Internet Information Services view.

IOA-000055894

When you select a subnode in the Releases view and then build a release, the focus now stays on the subnode that you had selected. Previously in some project types, the focus shifted to the root node. In other project types, the right pane switched to the inline help, and the focus was removed from the selected subnode.

IOA-000055986 (Basic MSI)

When you upgrade an Express project to the Premier or Professional editions of InstallShield, InstallShield now adds 64-bit folder locations to the list of predefined folders that you can show or hide in the "Destination computer's folders" pane of the Files and Folders view. The list of predefined folders is displayed when you right-click a node in the "Destination computer's folders" pane and then click Show Predefined Folder.

IOA-000056024 (Basic MSI)

If your installation includes billboards that you configured in the Billboards view, the progress bar now starts moving as soon as it is displayed on the progress dialog. Previously, the progress bar did not start moving until the installation began copying files to the target system.

IOA-000056027 (Basic MSI, InstallScript MSI)

If you add a new entry to the Directory table of your project and you type a value for the DefaultDir field, InstallShield no longer automatically creates a new string identifier and associates it with the DefaultDir field. This behavior is consistent with InstallShield 2009 and earlier.

IOA-000056034 (Basic MSI, InstallScript, InstallScript MSI, InstallScript Object, Merge Module, Transform)

You can once again use CTRL+C and CTRL+V to copy and paste files from one component to another in the following areas:

- For Basic MSI, InstallScript MSI, Merge Module, Transform projects: In the Files subview within the Components and Setup Design views
- For InstallScript and InstallScript Object projects: In the Static File Links subview within the Components and Setup Design views

Using CTRL+C and CTRL+V in these areas of InstallShield 2010 had no effect.

IOA-000056062 (Basic MSI, InstallScript MSI, Merge Module)

If you clone a managed-code custom action, the value of the In-Script Execution setting that was configured for the original custom action is now used in the cloned custom action. Previously, Immediate Execution was always selected for the In-Script Execution setting of the cloned custom action, regardless of the value for the original custom action.

IOA-000056205 (Basic MSI, InstallScript MSI)

The InstallShield Prerequisites panel of the Release Wizard now has a "Follow individual selections" option. Therefore, this panel now has the same options that are available from the InstallShield Prerequisites Location setting, which is displayed on the Setup.exe tab when you select a release in the Releases view.

IOA-000056256 (Basic MSI, InstallScript, InstallScript MSI, InstallScript Object, Merge Module)

If you are using the XML file format for your project file (.ism) and you use the Direct Editor view to manually add a new entry to the Binary table, InstallShield now lets you reopen the project file. Previously, if you added a new Binary table entry through this view, saved and closed the project, and then tried to reopen the project, InstallShield displayed a message explaining that the .ism file could not be opened.

IOA-000056376 (InstallScript Debugger)

You can now use the InstallScript Debugger to debug InstallScript if the drive letter where the original .dbg file was created does not exist on the debug machine.

IOA-000056406

When you rebuild an existing release, InstallShield no longer changes the modified date of the build log files and report files that were created during previous builds.

IOA-000056424

In the InstallShield Help Library, the "GetAndAddAllFilesCost " help topic now contains the correct syntax for the InstallScript function GetAndAddAllFilesCost:

```
GetAndAddAllFilesCost (szSrcDir, szWildcard, szTargetDir, nClusterSize,
nvInstallCostHigh, nvInstallCostLow, nvUninstallCost);
```

Previously, some of the parameters were listed in the wrong order.

IOA-000056470 (InstallScript, InstallScript MSI)

The text on the Japanese welcome dialog has been updated to reflect the current default text on the English version of this dialog.

IOA-000056534 (InstallScript MSI)

If you create an InstallScript MSI installation that uses the InstallScript engine as an embedded UI handler (that is, the InstallScript User Interface Type setting in the General Information view is set to the New Style (Requires Windows Installer 4.5) option)), the product can now be removed through Add or Remove Programs on Windows Vista and later systems. Previously, the only way to remove the product from these systems was to run Setup.exe again and select the Remove option. Attempting to remove the product through Add or Remove Programs did not uninstall the product.

IOA-000056536 (InstallScript)

You can once again enter a script-defined folder location such as the following one in the Destination setting for a component:

<FOLDER_COMMON_APP>\<IFX_COMPANY_NAME>\<IFX_PRODUCT_NAME>

Previously, if you tried to enter a such a path in this setting, InstallShield displayed an error stating that your entry contained invalid characters.

IOA-000056568 (Basic MSI, InstallScript, InstallScript MSI)

For Basic MSI and InstallScript MSI projects, specifying a value in the Limit Interval (minutes) setting for an application pool in the Internet Information Services view no longer causes a run-time error on systems that have IIS 7 or later. If you specify a value in this setting in an InstallScript project, the application pool is now installed at run time; previously, the application pool was not installed, and no error was reported.

IOA-000056624

Step 1c in the "Configuring the TCP Port and Site Numbers" help topic in the InstallShield Help Library has been corrected: The property name is no longer enclosed within square brackets, since the brackets are not used in the Property Manager view.

IOA-000056730 (InstallScript)

If your machine already has one or more files in C:\Reports when you build an InstallScript installation, build error -5046 no longer occurs.

IOA-000056769

The "Connecting to an Instance of Oracle and Running SQL Scripts" help topic in the InstallShield Help Library has been updated to include instructions on how to prepare an InstallShield prerequisite for version 11.1.07 of the Oracle 11g Instant Client. This help topic also explains what to do if you are using a version of Oracle that was released after 11.1.0.7. In this case, it is necessary to open the Oracle 11g Instant Client 11.1.0.7 prerequisite in the InstallShield Prerequisite Editor and rename the prerequisite to reflect the appropriate version number. In addition, it is also necessary to update the prerequisite's conditions to reflect the appropriate version number.

IOB-000050440

If you use MSBuild to build a Visual Studio solution that includes an InstallShield project, and if the InstallShield project contains the primary output, build warning -6248 no longer occurs in certain scenarios. Previously, this warning erroneously occurred in some cases.

IOB-000053545 (Basic MSI, InstallScript MSI)

If you have a Visual Studio solution that includes an InstallShield project, InstallShield can now list dependencies for primary outputs when an output group has more than one folder. Previously, InstallShield did not include support for viewing dependencies if an output group had more than one folder.

IOB-000053816

The "DialogSetInfo" help topic in the InstallShield Help Library now lists the project types that include support for the InstallScript function DialogSetInfo:

- InstallScript
- InstallScript MSI (in event-driven InstallScript—not in InstallScript custom actions)

Previously, this help topic did not list any project-specific information.

IOB-000056986

The "Specifying Dialog Skins" and "Dialog Skins" help topics in the InstallShield Help Library now state that a dialog skin cannot be applied to or removed from a standard dialog in an InstallScript or InstallScript MSI project after

that dialog has been edited in the Dialog Editor. Therefore, if you want to specify skins for dialogs that you plan on modifying, you must specify the skin first, and then edit it.

IOB-000058418 (InstallScript)

If an InstallShield prerequisite needs to be launched during upgrades or during maintenance mode of an InstallScript release, it is now launched. Previously, the only time that InstallShield prerequisites were launched was during a first-time installation, if the prerequisite needed to be installed on the target system.

IOB-000058716

The offline activation process for InstallShield now works as expected on Windows 7 systems. Previously in some cases, the activation dialogs indicated that the activation was successful before it had been completed (after the request code was saved, but before the response code could be entered), and successive launches of InstallShield displayed the activation dialogs again.

IOB-000058755 (InstallScript)

If you use the Internet Information Services view to configure an IIS virtual directory in an InstallScript project that also contains one or more merge modules, the virtual directory and the merge modules are installed at run time. Previously, the merge modules were installed at run time, but the virtual directory was not.

IOB-000058888

If you use the search-and-relace functionality in the Direct Editor view and the "Match whole word only" check box is selected, InstallShield now replaces only whole-word matches. Previously in some cases, matches in partial words were replaced.

IOB-000059291 (Basic MSI, InstallScript MSI)

If you use a Windows Installer property to set the TCP port number at run time for an IIS Web site that is configured in the Internet Information Services view, the Web site is now uninstalled when the IIS Web site's component is uninstalled, provided that the Web site's component is not permanent. Previously in this case, the Web site could not be uninstalled.

IOB-000059388 (Basic MSI, InstallScript, InstallScript MSI)

If you configure some of the virtual directory settings in the Internet Information Services view but you leave the default value of No for the Directory Browsing setting, the resulting installation no longer turns on directory browsing for the virtual directory on the target system.

IOB-000059398 (InstallScript, InstallScript MSI)

The InstallScript function FeatureFileInfo now returns an error if the specified file is not found on the target system. Previously, the function always returned the number 0, indicating that the function was successful, even if it was not successful.

IOC-000040911 (Basic MSI, InstallScript MSI, Smart Device)

If you enter a comma in the Application Name setting or the Company Name setting on the Application Information panel of the Windows Mobile Wizard or the Smart Device Setup Wizard, and then you click Next, the wizard displays an error message that explains that commas cannot be used. Previously, no such validation was performed, even though a comma is not a valid character for these settings. Thus, a run-time error occurred if you used a comma for these settings; the run-time error indicated that the application could not be installed because of an invalid setup file.

IOC-000045115

The "OnComponentError" help topic in the InstallShield Help Library now has information on the following parameters: ErrorInfo.FileError.Description, ErrorInfo.FileError.File, and ErrorInfo.FileGroup. Previously, these parameters were not documented.

IOC-000050558

The "SdOptionsButtons" help topic in the InstallShield Help Library now has an important note that indicates that the control identifiers of the push button controls on the SdOptionsButtons dialog should not be changed, and that these push button controls should not be set as the default control. This help topic also states that if you change the control identifiers or set one of the these controls as the default control of the dialog, the SdOptionsButtons dialog will not work as expected.

IOC-000055567 (Basic MSI, InstallScript MSI)

If the InstallValidate and InstallInitialize actions in your project are sequenced consecutively, one right after another, with no unused sequence numbers in between, InstallShield now correctly resequences the RemoveExistingProducts action when you switch from the "Install setup then remove unneeded files" option to the "Completely uninstall old setup before installing new setup" option in the Upgrades view. Previously in this scenario, InstallShield was not able to resequence the RemoveExistingProducts action.

IOC-000056288 (Basic MSI)

If you rename a dialog that is included in the InstallUISequence table of your project, InstallShield now renames the reference to that dialog in the InstallUISequence table accordingly.

IOC-000056960

In the "DoInstall Example" help topic of the InstallShield Help Library, the sample code has been corrected. Previously, it used outdated return values.

IOC-000060340 (InstallScript)

If an installation installs a file from a shared component onto a target system, the original security settings for that file are no longer modified on the target system. Previously, if the file was locked, the security settings for that file were changed on the target system.

IOC-000061176

In the InstallShield Help Library, the "Response File Silent Header" help topic has been corrected. This help topic states that when you are creating response files, you should use the same version of InstallShield to create the response file that will be used to run the silent installation.

IOC-000066525

In the InstallShield Help Library, the "InstallScript Functions that Are Logged for Uninstallation" help topic has been corrected.

IOC-000071368 (Basic MSI, InstallScript MSI, QuickPatch)

The Target Product Name setting on the Identification tab for a patch configuration in the Patch Design view now permits the use of apostrophes; this applies to Basic MSI and InstallScript MSI projects.

The Target Product Name setting on the Identification tab of the Build Settings area in the General Information view also permits the use of apostrophes; this applies to QuickPatch projects.

Previously, apostrophes were not permitted because earlier versions of the Microsoft tool Patchwiz.dll did not support them.

IOC-000074259 (InstallScript)

If the Visual Basic Runtime object is included in a project and the installation installs the Visual Basic Runtime on a Japanese system, the status text is no longer garbled.

IOC-000075959 (Basic MSI, InstallScript MSI, Merge Module, Transform)

If you use CTRL+C and CTRL+V to copy and paste shortcuts from one component to another, the changes are now retained when you save your project. Previously, the shortcut was not moved from one component to another.

IOC-000076034 (QuickPatch)

If you clear the check box for a custom action in the Custom Action area of the General Information view of a QuickPatch project and the custom action in the base installation project had a condition that used the OR operator, the custom action no longer runs when the QuickPatch package is run on a target system. Previously in some cases, the custom action was run.

IOC-000077506

In the InstallShield Help Library, the "Migrating from InstallShield Professional 6.x" help topic now explains what to do if you upgrade an InstallShield Professional 6 project to InstallShield 2011 and a message such as the following one is displayed at run time: "The installed version of the application could not be determined. The setup will now terminate."

IOC-000078276 (Basic MSI)

If you clone a dialog that contains a radio button group control and then you change the property that is associated with the radio button group control on the cloned dialog, InstallShield no longer removes the radio button controls that are on the original dialog.

IOC-000079459 (Basic MSI, InstallScript MSI)

If you use the patch optimization functionality when creating an upgrade, InstallShield now synchronizes registry keys in components that use COM extraction. Previously, such registry data was not synchronized.

IOC-000079640

The InstallShield Help Library contains a new help topic called "Customizing File Properties for the Setup Launcher." This help topic explains which settings in InstallShield are used to configure the version resources of the Setup.exe setup launcher. The inline help that is displayed in the Releases view when you click various settings such as the Use Custom Version Properties setting and the Launcher Copyright setting now contains a hyperlink to the new help library topic.

IOC-000079814 (Basic MSI, InstallScript MSI)

If an installation includes the Oracle Instant Client prerequisite and launches it at run time, and if the installation tries to connect to an Oracle database server using a TNS service name, the installation can now connect to the server without rebooting after the Oracle Instant Client prerequisite is installed.

IOC-000080799 (Basic MSI, InstallScript, InstallScript MSI)

If you edit the QueryDatabasesCmd field in the ISSQLDbMetaData of your InstallShield project to create a custom SQL query that returns multiple record sets on a target system, all of the appropriate databases are listed when end users try to browse for a database from the SQLLogin dialog. Previously, only the results of the first record set were listed.

IOC-000080835 (Basic MSI, InstallScript, InstallScript MSI)

If you create an installation that uses the logon information functionality (which is documented in the "Adding the Ability to Create or Set an Existing User Account" help topic) and end users run the installation on a Japanese system, end users no longer encounter issues when entering certain passwords. Previously in some cases, if end users entered certain passwords, the user account was not created at run time.

IOC-000080886 (Basic MSI, InstallScript MSI)

If you specify a conditional statement for a SQL script that is scheduled to be run during rollback, the SQL script is now run during rollback only when the condition evaluates as true. Previously, the SQL script was run during rollback regardless of whether the condition was true.

IOC-000080964

The full Visual C++ 9.0 merge modules are no longer displayed in the Redistributables view and the Objects view as Visual C++ 9.0 policy merge modules; in addition, the policy merge modules are no longer displayed as full merge modules.

IOC-000081031

When you save an InstallShield project file (.ism), the "File is ready for archiving" check box is now selected by default; this check box is an advanced attribute that is available on the .ism file's Properties dialog box.

IOC-000081136 (Basic MSI, InstallScript MSI, Merge Module, Transform)

If you use the Import REG File Wizard to import a .reg file that contains square brackets ([]) or curly braces ({ }), InstallShield now includes these special characters in the corresponding Registry table entries. Note that the .reg file that you are importing should not include escape characters for the special characters. Previously, InstallShield did not import the registry data that included the special characters.

IOC-000081641 (InstallScript)

If you specify a value in the Display Icon Index setting of the General Information view of an InstallScript project, the installation now uses the appropriate icon resource for your product's entry in Add or Remove Programs. Previously, the installation always used the first icon resource of the specified icon file.

IOC-000081714 (Basic MSI)

If you upgrade a project that was created in InstallShield Express 2 to InstallShield 2011 and open the Text File Changes view, Windows Installer error 2228 no longer occurs.

IOC-000081718 (InstallScript, InstallScript MSI)

If you edit an InstallScript dialog, and then you change your machine's DPI value and build the installation that contains that edited dialog, the size of the dialog is now correct at run time. Previously, the edited dialog was a different size than the unedited dialogs at run time.

IOC-000081856 (InstallScript, InstallScript MSI)

The default code for the InstallScript event OnCustomizeUninstInfo and the InstallScript SdCustomerInformation* functions no longer contains any Windows 9x–specific code.

IOC-000082083

If you have a Visual Studio setup project that uses a Windows Installer property for the Default Location setting, you can use InstallShield to convert that Visual Studio project to an InstallShield Basic MSI project. Previously, fatal error -9021 occurred if you tried to convert this type of Visual Studio setup project.

IOC-000082134 (InstallScript)

If you build a release for a project that contains an InstallScript Object and then you rebuild using the Refresh Build command on the Build menu, error -9008 no longer occurs.

IOC-000082208 (Basic MSI, InstallScript MSI, QuickPatch)

InstallShield now lets you create a standard patch (through the Patch Design view) or a QuickPatch package that includes more than 32,767 files. Previously in this case, build error -6415 occurred.

IOC-000082424 (Basic MSI)

The words "Arabic" and "Hebrew" are now translated in the language .ini files for every supported language. These translations are used on the language selection dialog.

IOC-000082530 (InstallScript)

If your machine already has one or more files in C:\LogFiles when you build an InstallScript installation, build error -5045 no longer occurs.

IOC-000082646

In the InstallShield Help Library, the "InstallShield Custom Action Reference" help topic now contains details about the ISSQLServerInitialize and ISSQLQueryDatabases custom actions.

IOC-000084101 (Basic MSI)

The IDS_PREVENT_DOWNGRADE_EXIT string, which is displayed on a message box at run time if an end user tries to install an old version of the product over a new version, now contains a translated version of the word "OK" for languages in which the OK button on the message box is translated.

IOC-000084807 (Basic MSI, InstallScript, InstallScript MSI)

If you use INSTALLSHIELD_IIS_NEXT_NEW_SITE_NUMBER in your project to install an IIS Web site to the next available new site number, and if the target system already has site numbers 9 and 10, the installation now installs the Web site to site number 11. Previously on systems that had IIS 7 or later, the installation failed.

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 6

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 Q208912.