



InstallShield 2012

Release Notes

originally released August 2011; updated to include SP1, released November 2011

Introduction

InstallShield is the industry standard for authoring high-quality Windows Installer- and InstallScript-based installations, as well as Microsoft App-V packages. InstallShield 2012 offers powerful, new features and enhancements that make it easy to use the latest technologies, as well as enable you to be more agile, flexible, and collaborative when developing installations.

The new Suite installation support in InstallShield 2012 offers flexibility for developing, packaging, and distributing products. With this new functionality, you can bundle one or more product installations into a single, unified installation, eliminating the need to develop a complex custom launcher or bootstrap application. Suite installations use an entirely new end-user interface, with fresh, modern wizard pages that you can customize as needed. For end users who download your product from the Web, you can create a small Suite Setup.exe file that downloads and runs only the required packages that needs to be installed.

The redesigned installation collaboration feature in InstallShield 2012 supports distributed, collaborative installation development by enabling development teams to create and manage their discrete portions of the installations simultaneously. This capability lets release engineers reuse those discrete portions within multiple installations, and it improves the efficiency of development teams.

InstallShield 2012 now has support for building ISO/IEC 19770-2 software identification tags and including them in installations. This functionality gives your enterprise customers better information for software asset management and license optimization initiatives.

Other features and enhancements in InstallShield 2012 include new InstallShield prerequisites that support the latest technologies, improved COM extraction, and support for 64-bit dependency scanning.

For the latest information about InstallShield 2012, including updates to these release notes, see Knowledge Base article [Q211161](#).

InstallShield 2012 Service Pack 1 includes improvements and resolves issues.

Changes in SP1 (November 2011)

To obtain SP1, see KB article [Q201298](#).

Support for Digitally Signing Software Identification Tags

If you configure your project to include a software identification tag and you also configure the release in the Releases view to use a .pfx file to digitally sign your release, InstallShield digitally signs the tag at build time. Note that the .NET Framework 2.0 or later must be installed on your build machine in order to sign a tag file.

IOA-000065046

The conditions for the InstallShield prerequisite that installs Crystal Reports Basic for Visual Studio 2008 (x64) have been fixed. Previously, they were incorrect, and the installation did not run on x64 target systems when needed.

IOA-000065062 (Basic MSI, DIM, InstallScript MSI, Merge Module)

Log messages that are written by the built-in InstallShield custom action ISSearchReplaceUninstall now accurately indicate what is occurring during uninstall. Previously, the log indicated that the component that contained the text file changes was not being removed, even if it were being removed.

IOA-000065241 (InstallScript)

It is now possible to add an InstallScript Object to a feature in an InstallScript project. Previously, two of the InstallShield DLL files were not registered properly during the installation; therefore, adding an InstallScript Object to a feature in an InstallScript project failed, and it caused an error ("The object <ObjectNamem> could not be inserted. You may need to reinstall the object in order to use it.").

IOA-000065300 (Suite)

Selecting a non-.ico file for the icon of a Suite Setup.exe file no longer causes build warning -7212, and it no longer uses the default icon instead of the selected custom icon.

IOA-000065347, IOA-000065397 (Basic MSI, InstallScript MSI)

If you use the Release Wizard to create the Web type of release, the settings on the Setup.exe tab are no longer disabled. The Setup Launcher setting on this tab is now always set to Yes; No is no longer available, since creating a Web release requires the Setup.exe setup launcher.

IOA-000065378 (Basic MSI, InstallScript MSI)

The dependency scanners in InstallShield now add the appropriate 32-bit merge module for a 32-bit managed DLL that has native dependency. Previously in some cases, InstallShield added a merge module that had the appropriate name but that was targeting 64-bit instead of 32-bit platforms.

IOA-000065380 (Suite)

The "Suite Setup.exe Command-Line Parameters" topic in the InstallShield Help Library has been corrected. For example, previously, this help topic listed /I instead of /language.

IOA-000065382 (Suite)

The inline help for the command line settings and the silent command Line settings in the Packages view have been expanded. The documentation now explains that if you include an .msi or .msp package in a Suite installation, you should specify only Windows Installer properties in these settings.

IOA-000065386 (Basic MSI, DIM, InstallScript MSI, Merge Module)

Build warning -6637, which warns about invalid registry data, no longer occurs when you build a release that includes a .NET file whose REG File to Merge at Build feature setting specifies a .reg file that contains a multistring value.

IOA-000065468

You can now import registry files (.reg) that contain UNIX or Mac style line endings into InstallShield projects.

IOA-000065478 (Suite)

When you add a 64-bit .msi package to a Suite project, InstallShield now properly detects that the package is 64 bit, enabling a package to install to 64-bit locations. Previously, the 64-bit locations may have been redirected to 32-bit locations.

IOA-000065590 (Suite)

At run time, if a child feature is selected by default (that is, if the condition that is configured in the child feature's Condition setting evaluates as false on the target system), and an end user selects that child feature in the InstallationFeatures wizard page, its parent feature is now also selected. A child feature cannot be selected without its parent feature being selected. When the installation is done, both the parent and child features are present on the target system. Previously, even though the run-time UI showed that the parent feature was selected, it was not installed, and neither was the child feature.

IOA-000065625 (Basic MSI, DIM, InstallScript MSI, Merge Module)

Self-unregistration of .exe files is now performed correctly. Previously, self-unregistration failed, and a run-time error was displayed.

IOA-000065703 (Suite)

If you try to add a file that is already associated with a package in the Packages view, the Resolve Conflict dialog box opens. If you specify that you want to replace the existing file, InstallShield now replaces the file. Previously, the file was not replaced.

IOA-000065850 (Suite)

It is now possible to set and then unset the Anchor setting for a control in the Wizard Interface view. Previously, once a value had been added to the Anchor setting, it was no longer possible to revert back to a blank setting.

IOA-000065889 (Suite)

If you change a brush style from a solid brush to a gradient brush, the style is applied correctly to areas of the wizard pages. Previously, the gradient settings were not properly stored in the XML, and this led to display problems.

IOA-000065947 (Suite)

Each combo box control and list box control in the Suite user interface has a new Content Property setting. This setting lets you specify the options that you want to be listed in the control. It also lets you assign a different property value to each option. At run time, the installation sets the property that is identified in the control's Property setting with the value that corresponds with the option that the end user selects.

New Features in InstallShield 2012 Original Release Version (August 2011)

Ability to Create Suite Installations that Run Multiple Packages; New Modern, Customizable End-User Interface; Ability to Build Hybrid 32-Bit/64-Bit Installations

The Premier edition of InstallShield now lets you build a Suite installation that uses the next-generation setup launcher (Setup.exe) to conditionally run multiple installations and apply Windows Installer patches (.msp) as needed on target systems. This support is available through a new Suite project type. Suite installations can be run on systems with Windows XP and later and with Windows Server 2003 and later; they cannot be run on systems with Windows 2000. Windows Installer 3.1 is required if the Suite installation is installing a Windows Installer package.

Following are some highlights of this functionality.

Ability to Package Multiple Installations as a Single Installation

The new Suite project type contains a Packages view that lets you specify one or more of the following types of packages:

- Executable files (.exe), including Windows Installer–based and non-Windows Installer–based installations that you want to be run on target systems
- Windows Installer packages (.msi) that you want to be run on target systems
- Windows Installer patches (.msp) that you want to be applied on target systems

The Packages view also lets you include multiple .msi and .msp packages that you want to be run using transaction processing, a feature of Windows Installer 4.5 and later. The packages are chained together and processed as a single transaction. Each Suite installation can have multiple separate transactions. If one or more of the packages in a transaction cannot be installed successfully or if the end user cancels the installation, Windows Installer initiates rollback for all of the chained packages within the current transaction to restore the system to its earlier state.

The Suite installation launches the appropriate packages at run time based on conditions that you have defined and the order in which you listed the packages in the Packages view.

Modern, Customizable User Interface for the Installation; New Editor for Customizing Suite Setup.exe Wizard Pages

The Suite project type in InstallShield includes an entirely new end-user interface, with redesigned, modern built-in wizard pages that you can include and customize in your Suite installations. The new wizard page editor in this project type lets you add, sequence, and remove pages as needed; it also lets you modify the layout of any page—adding, moving, and removing a variety of different kinds of controls.

Support for Combining 64-Bit and 32-Bit Windows Installer Packages Into a Single Installation

As more and more users move to 64-bit versions of Windows, you may need to now, or in the near future, deliver to customers a single installation that installs to 32-bit locations on 32-bit systems and to 64-bit locations on 64-bit systems. The Suite project type lets you include both 32-bit packages and 64-bit packages in one Suite installation, and run only the appropriate packages on each target system. Previously, other alternatives required delivering two separate installations (one for 32-bit systems and one for 64-bit systems) or creating a custom launcher, a bootstrap application, or an InstallScript installation.

Support for Displaying a Single Progress Bar that Shows the Overall Status of the Entire Suite of Packages

The progress bar on the progress wizard page in a Suite installation shows the status of the entire suite of packages. This integrated progress bar presents end users with a clear visual indication of how far along the Suite installation is overall. To ensure that end users see only the integrated progress bar, you must include only .exe installations for which you have specified command-line parameters that hide the user interface (that is, run silently), .msi packages, and .msp patches.

Optional Add or Remove Programs Entry for the Suite Installation

Suite projects let you specify whether you want to have an entry in Add or Remove Programs for your Suite installation. This entry lets end users perform maintenance for your Suite, modifying or removing if needed. The General Information view in a Suite project has a Show Add or Remove Programs Entry setting that lets you indicate the appropriate behavior.

If you want to show only a single entry for the entire Suite, ensure that you hide the entries from the packages that you include in the Suite project.

Support for Running Suite Installations Without a User Interface

End users can run your Suite installations with a user interface or silently, without a user interface. Silent installations run without user intervention; end users can avoid monitoring the installation and providing input through run-time wizard pages.

Default Setup.exe User Interface Strings Included for All 35 Supported Languages; Ability to Edit the Suite Run-Time Strings

Translations of all of the default strings that are displayed in the built-in wizard pages of Suite projects are available in all 35 of the run-time languages that InstallShield supports. All of these Suite strings are displayed in the String Editor view of a Suite project. This String Editor view offers the same robust support that other types of projects offer, giving you complete and centralized control over the text strings that are displayed at run time during the Suite installation process.

Small Base Setup.exe File with the Ability to Download Only the Required Packages As Needed

Suite projects include flexible options for specifying the run-time source location of each package in the Suite installation. When you define the packages that you are including in a Suite project, you can specify the location of each individual package. The available options are:

- On the Web, available for download by Setup.exe if needed
- Embedded in Setup.exe and extracted to the target system if needed
- Uncompressed and stored on the Suite source media

The base Setup.exe file that is used for Suite installations is much smaller than the base Setup.exe files that are used for Basic MSI, InstallScript MSI, and InstallScript installations. Thus, your end users can quickly download a small Suite Setup.exe file, and the Setup.exe file can download and launch one or more required packages as needed.

This feature resolves the following issue: IOA-000055518.

Redesigned, Expanded Support for Modularizing Installation Projects to Enable Reuse and Distribute Development Work

InstallShield includes a new project type called DIM, which is known as a *developer installation manifest*. A DIM project is a feature-sized collection of related items such as product files, shortcuts, registry entries, text file changes, IIS Web sites, and other elements that together make up a discrete portion of a product installation. Some benefits of using DIMs are as follows:

- DIMs include support for virtually the same functionality that is available in Basic MSI projects. This gives authors of DIMs all of the flexibility that they need to develop their portions of an installation.
- Release engineers can reuse DIMs in multiple Basic MSI projects, enabling efficiency.
- Working with DIMs enables multiple team members to contribute to the development of the installation simultaneously. Each software developer or other team member can work on a separate DIM that the release engineer can reference in one or more Basic MSI projects.

Once you have created a DIM, you can add it to a Basic MSI project in one of two ways:

- By reference—You can add a reference for a DIM project to your Basic MSI project through the Setup Design view or the DIM References view. With this method, the DIM elements are merged into the Basic MSI project

at build time. Each time that you build the Basic MSI installation, InstallShield references the latest version of the DIM project and includes it in the installation that it generates. This method is the more commonly performed method.

- By import—You can import a DIM project into your Basic MSI project by using the new Import DIM Wizard. This method is a one-time, irreversible import that merges the DIM data into the Basic MSI project at design time.

This redesigned, expanded DIM support replaces the previous support that required users to create DIMs in a separate tool called InstallShield Collaboration, and then import the DIM files into InstallShield Basic MSI projects. The new DIM support is more robust than the previous support. The new DIM project type lets you have virtually the same complete level of authoring flexibility that is available for Basic MSI projects. For example, the new DIM project type lets you have full control over component creation: you can add components to a new DIM project, set the key file of a component, and configure the component's settings. The new DIM project type also lets you configure IIS Web sites. InstallShield Collaboration did not have that sort of flexibility over component design, and it did not have any built-in support for configuring IIS Web sites.

The ability to create DIM projects is available in the Premier edition of InstallShield. This support is also available in the InstallShield Developer Installation Manifest Editor, a new collaboration add-on. The ability to add DIM files to Basic MSI projects is available in the Premier edition of InstallShield.

New InstallShield Prerequisites for Internet Explorer 9, SQL Server 2008 R2 Native Client, Windows Identity Foundation, and Other Redistributables

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

- Internet Explorer 9.0
- Microsoft SQL Server 2008 R2 Native Client 10.50.1600.1
- Windows Identity Foundation
- Microsoft VSTO 2010 Runtime (x64)
- Microsoft Office 2010 PIA (This prerequisite installs the Microsoft Office 2010 Primary Interop Assemblies. To use this prerequisite, download the PrimaryInteropAssembly.exe file from Microsoft's Web site and run it to extract the .msi file.)

This feature resolves the following issues: IOA-000056889, IOA-000058423, IOA-000056974, IOA-000061523, IOA-000064084, IOA-000060652, IOA-000062254.

Support for 64-Bit Dependency Scanning

The Static Scanning Wizard and the Dynamic Scanning Wizard—dependency scanners in InstallShield—now include support for identifying 64-bit dependencies of the 64-bit files in your project. If you are using InstallShield on a 64-bit version of Windows Vista or later or a 64-bit version of Windows Server 2008 or later, the scanners can detect the 64-bit dependencies. The wizards let you specify whether you want to include each detected potential dependency in your project.

In addition, if you use InstallShield on a 64-bit version of Windows Vista or later or a 64-bit version of Windows Server 2008 or later, and you use either of the following built-in methods for detecting dependencies, InstallShield can scan for 64-bit dependencies of the 64-bit .NET assemblies in your project:

- The Static Scanning Wizard helps you identify a 64-bit .NET assembly's possible dependencies on demand. This wizard displays a list of the dependencies that it finds, and it lets you specify whether you want to include each one in your project.

- A component's .NET Scan at Build setting lets you specify whether you want InstallShield to identify a 64-bit .NET assembly's dependencies each time that you build your project. For InstallScript projects, the component's .NET Assembly setting must also be set to Local Assembly. If InstallShield detects any possible missing dependencies at build time, InstallShield incorporates them into the release that it generates.

InstallShield must be installed on a 64-bit operating system in order to scan 64-bit files for 64-bit dependencies. Note that if you use InstallShield on a 32-bit version of Windows, these built-in scans can check for only 32-bit dependencies of the 32-bit files in your project. If your project includes 64-bit files, you can manually add any dependencies to the project as needed.

Support for Setting Permissions for Files, Folders, and Registry Keys in 64-Bit Locations

InstallShield has support for setting permissions for files, folders, and registry keys in 64-bit locations. The support varies, depending on which project type you are using.

Using the Custom InstallShield Handling Method in Windows Installer-Based Projects

If you are using the custom InstallShield handling method to set permissions for files, folders, and registry keys, you can now set permissions for these items when they are in 64-bit locations; this includes 64-bit locations in the registry, as well as the 64-bit System32 folder on 64-bit systems. The file, folder, or registry key must be included in a component that is marked as 64 bit (that is, Yes is selected for its 64-Bit Component setting). Previously, if the component was marked as 64 bit, permissions could not be set, and a run-time error was displayed.

This custom InstallShield handling support is available in the following project types: Basic MSI, InstallScript MSI, Merge Module, MSI Database, MSM Database, and Transform. The Locked-Down Permission setting in the General Information view lets you specify which method you want to use for setting permissions: either the custom InstallShield handling method or the traditional Windows Installer handling method.

Using the InstallScript Function SetObjectPermissions in InstallScript-Based Projects and Windows Installer-Based Projects

If the REGDB_OPTION_WOW64_64KEY option is enabled and you use the InstallScript function SetObjectPermissions to set permissions for a 64-bit registry key, the function can set its permissions correctly. To force SetObjectPermissions to set permissions for a 64-bit registry key regardless of whether the REGDB_OPTION_WOW64_64KEY option is enabled, you can use the new IS_PERMISSIONS_OPTION_64BIT_OBJECT constant; pass this constant in the nOptions parameter of SetObjectPermissions. Note that the IS_PERMISSIONS_OPTION_64BIT_OBJECT constant should not be passed on 32-bit target systems.

If file system redirection is disabled using the WOW64FSREDIRECTION constant when SetObjectPermissions is called to set permissions for a file or folder in the 64-bit System32 folder, the function can set the permissions correctly. If file system redirection is not disabled, the permissions cannot be set.

You can use the SetObjectPermissions function in InstallScript events in InstallScript and InstallScript MSI projects. You can also use this function in InstallScript custom actions in the following project types: InstallScript, Basic MSI, InstallScript MSI, and Merge Module.

This feature resolves issue IOA-000056378.

Improvements for COM Extraction

InstallShield supports a new monitoring method for COM extraction. If you are using InstallShield on a Windows Vista or later system or a Windows Server 2008 or later system, this new method is used by default. The method uses a kernel driver to monitor the areas of the registry that are modified during dynamic COM extraction at build

time and static COM extraction at design time. It combines the advantages that the earlier methods provided, allowing the DLL to read existing registries entries and preventing changes to the build machine.

If necessary, you can switch between the three different COM extraction methods by setting the value data of the UseAPIRegistryHooks registry value, which is in the registry key HKEY_LOCAL_MACHINE\SOFTWARE\InstallShield\RegSpy (on 32-bit machines) or HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\InstallShield\RegSpy (on 64-bit machines). Possible REG_DWORD value data are:

- **0**—Use API hooking to read existing registry entries for the DLL.
- **1**—Use registry redirection to prevent making changes to the registered DLLs on the build machine. If the value is not set, this is the default behavior on Windows XP and Windows Server 2003 systems.
- **2**—Use the new kernel mode monitoring, which combines the advantages of both of the other methods. If the value is not set, this is the default behavior on Windows Vista and later systems and on Windows Server 2008 and later systems.

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

Predefined System Searches for Adobe Reader 10, Internet Explorer 9, and Microsoft Office

InstallShield has new predefined system searches:

- Adobe Reader 10
- Internet Explorer 9
- Microsoft Office 2010
- Microsoft Office 2007
- Microsoft Office 2003

If your installation requires one or more 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.

This feature resolves the following issues: IOA-000055355, IOA-000062254, IOA-000062255.

Support for Software Identification Tagging

ISO/IEC 19770-2 is an international standard for the creation of software identification tags. A software identification tag is an XML-based file that contains descriptive information about the software, such as the product name, product edition, product version, and publisher. Software asset management tools collect the data in the tags to provide accurate application identification for software that is installed in an enterprise.

Software identification tagging is evolving as an industry standard, enabling independent software vendors to create smarter applications that give their customers better information for software asset management and license optimization initiatives. Including the identification tag in your product's installation makes it possible for your customers to use tools that can monitor their internal usage of your product, allowing them to manage and optimize the number of licenses of your product that they obtain from you, and stay in compliance with your licensing policies.

InstallShield includes several new settings in the General Information view that let you specify information that is required to create an identification tag for your product. This view also contains a new Use Software Identification

Tag setting that lets you specify whether you want InstallShield to create the tag at build time and include it in your installation; the default value of this setting is Yes.

Note that if Yes is selected for the Use Software Identification Tag setting but you have not entered values in one or more of the required identification settings (the Unique ID, Tag Creator, and Tag Creator ID settings in the General Information view), build warning -7235 occurs, once for each of the settings that are blank. This build warning explains that the software identification tag could not be created and included in the installation because a specific required setting was left blank. To resolve this warning, enter appropriate value in each specific setting, or select No for the Use Software Identification Tag setting.

The automation interface includes support for the new tag settings. The ISWiProject object includes a new EnableSwidtag property that lets you enable or disable the creation of a software identification tag in a project. It also includes a new SwidtagProperty property that you can use to configure various tag-related settings that are also configurable in the General Information view.

This feature applies to Basic MSI projects.

Enhancements in InstallShield 2012 Original Release Version (August 2011)

Merge Module Projects Now Include Built-In Support for IIS, Text File Changes, and XML File Changes

Several existing views are now available in Merge Module projects in InstallShield:

- Internet Information Services—This view enables you to create and manage new IIS Web sites, applications, virtual directories, application pools, and Web service extensions.
- Text File Changes—This view lets you configure search-and-replace behavior for content in text files—for example, .txt, .htm, .xml, .config, .ini, and .sql files—that you want to modify at run time on the target system.
- XML File Changes—This view lets you add references for nodes and node sets of XML files that you want to change at run time.

Use these views in Merge Module projects to configure IIS, specify text file changes, and specify XML file changes. When you build the merge module, add it to an installation project, and then build a release in the installation project, InstallShield includes the applicable run-time support in the installation.

Previously, these views were available only in installation projects.

This enhancement resolves issue IOC-000072712.

New Application Pool Identity Option for IIS 7.x

A new ApplicationPoolIdentity option is available in the Identity setting for an application pool that is configured in the Internet Information Services view. If you want a virtual identity account that is unique to the selected application pool to be used for running the application pool's worker process, select this option. Support for this option is available on target systems that have IIS 7 and later. If this new option is selected in an installation that is run on a system that has IIS 6, the NetworkService account is used instead for the application pool's identity.

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

This enhancement resolves issue IOA-000059870.

Automation Interface Enhancement: New RequiredExecutionLevel Property for Specifying the Required Execution Level for Setup.exe

The read-write property `RequiredExecutionLevel` has been added to the `ISWiRelease` object. This property corresponds with the Required Execution Level setting on the Setup.exe tab for a release in the Releases view. This property is available for Basic MSI, InstallScript, and InstallScript MSI projects.

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 Installations for InstallShield, InstallShield Add-Ons, and the Redistributable Files

You can obtain the installations of InstallShield, Standalone Build, and Repackager (which is available with the Premier edition of InstallShield) through either of the following methods:

- If you have the InstallShield DVD, the installations are on the DVD and you can find them using the DVD Browser.
- The InstallShield and Standalone Build installations are available for download as documented in the [InstallShield download and licensing instructions](#).

Additional installations—such as the redistributable files for the InstallShield prerequisites that are included in InstallShield, the .NET language pack prerequisite files (.prq), and InstallScript objects—are also available in those same locations.

The ability to create DIM projects is available in the Premier edition of InstallShield. This support is also available in the InstallShield Developer Installation Manifest (DIM) Editor. The DIM Editor is included on the InstallShield Premier DVD. It is also available for download from the same location as InstallShield, the Standalone Build, and Repackager.

Installing More than One Edition of InstallShield

Only one edition of InstallShield 2012—Premier, Professional, or Express—can be installed on a system at a time. In addition, the InstallShield 2012 DIM Editor cannot be installed on the same machine with any edition of InstallShield 2012.

Installing More than One Version of InstallShield

InstallShield 2012 can coexist on the same machine with other versions of InstallShield.

The InstallShield 2012 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.

Project Upgrade Alerts

The following information describes possible upgrade issues that may occur when you upgrade projects that were created with InstallShield 2011 and earlier to InstallShield 2012. It also alerts you to possible changes in behavior that you may notice between new InstallShield 2012 projects and projects that are upgraded from InstallShield 2011 or earlier to InstallShield 2012. For updates to this information, see Knowledge Base article [Q211163](#).

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

If you use InstallShield 2012 to open a project that was created with an earlier version, InstallShield 2012 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 .771 before converting it. Delete the .771 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 2012 projects in earlier versions of InstallShield.

You can upgrade projects that were created with the following versions of InstallShield to InstallShield 2012: InstallShield 2011 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 2012.

Build Warning -7235 for Basic MSI Projects

By default, software identification tagging is enabled in all Basic MSI projects. This applies to new projects that you create in InstallShield 2012, as well as projects that you have upgraded from earlier versions of InstallShield to InstallShield 2012.

If you build a release in a Basic MSI project without entering data in the required identification tag settings (the Unique ID, Tag Creator, and Tag Creator ID settings in the General Information view), and you leave tagging enabled in the project, build warning -7235 occurs. This build warning explains that the software identification tag could not be created and included in the installation because a specific required setting was left blank. To resolve this warning, enter an appropriate value in each specific setting, or select No for the Use Software Identification Tag setting in the General Information view.

COM Extraction Changes

InstallShield supports a new monitoring method for COM extraction. If you are using InstallShield on a Windows Vista or later system or a Windows Server 2008 or later system, this new method is used by default. The method uses a kernel driver to monitor the areas of the registry that are modified during dynamic COM extraction at build time and static COM extraction at design time. It combines the advantages that the earlier methods provided, allowing the DLL to read existing registries entries and preventing changes to the build machine.

If necessary, you can switch between the three different COM extraction methods by setting the value data of the UseAPIRegistryHooks registry value, which is in the registry key
HKEY_LOCAL_MACHINE\SOFTWARE\InstallShield\RegSpy (on 32-bit machines) or
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\InstallShield\RegSpy (on 64-bit machines). Possible REG_DWORD value data are:

- **0**—Use API hooking to read existing registry entries for the DLL.
- **1**—Use registry redirection to prevent making changes to the registered DLLs on the build machine. If the value is not set, this is the default behavior on Windows XP and Windows Server 2003 systems.

- **2**—Use the new kernel mode monitoring, which combines the advantages of both of the other methods. If the value is not set, this is the default behavior on Windows Vista and later systems and on Windows Server 2008 and later systems.

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

String Entries in Merge Module Projects

Each string identifier that is created in a Merge Module project now contains the Merge Module's module ID GUID. This applies to all new string identifiers that are created in all new Merge Module projects, as well as new string identifiers that are created in existing Merge Module projects that were upgraded from InstallShield 2011 or earlier to InstallShield 2012. Note that if you upgrade a Merge Module project from InstallShield 2011 or earlier to InstallShield 2012, the module ID GUID is not added to the string identifiers of any existing string entries.

The use of the GUID in string identifiers of Merge Module project string entries helps to minimize or eliminate conflicts that would occur if the same string identifier is used in a Merge Module project and also in an installation project that contains that Merge Module, and if different values are assigned to those two string identifiers.

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 2012, you can still do so. For instructions, see Knowledge Base article [Q200884](#).

Resolved Issues in InstallShield 2012 Original Release Version (August 2011)

1-173CAX (MSM Database)

The Open MSI/MSM Wizard no longer crashes when you try to open an .msm package that has an improperly formatted ModuleID in the ModuleSignature table.

IOA-000030237 (InstallScript, InstallScript MSI)

The exterior wizard panel dialogs, such as the SdWelcome dialog, now use a white background for any types of button controls, such as check box controls. The white background of the dialog controls matches the area on exterior dialogs that is white. Previously, the background of some of the controls was black.

IOA-000052170 (Basic MSI, InstallScript MSI)

If you use the Device Driver Wizard to configure a device driver as part of your installation, and if the .inf file refers to files in subfolders, InstallShield now creates the same corresponding folder structure and properly sets the destination for the driver files. Previously in this scenario, the structure was not created properly at run time, causing the installation to fail and roll back.

IOA-000053838 (Repackager)

The INSTALLDIR setting in the General Information view is now configured correctly when Repackager is used to repackage an installation and create a Basic MSI project for an INSTALLDIR Directory table entry in which the value of the DefaultDir column is something other than a dot.

IOA-000054204

Bookmark support has been added back to the script editors in InstallShield. The views that contain a 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). Bookmarks are markers that let you jump to specific lines within your script with a minimum number of keystrokes. Bookmarks are visible in the left margin of the script editor.

To add a new bookmark or clear an existing bookmark, place the insertion point in the applicable line of your script and then press ALT+K.

To move the insertion point to the next line that contains a bookmark, press CTRL+K.

To move the insertion point to the previous line that contains a bookmark, press CTRL+SHIFT+K.

IOA-000056626 (InstallScript)

If you use disk spanning when building an InstallScript release, InstallShield now uses the decimal part of the disk size when calculating the number of disks that are required. Previously, the decimal part of the value was not considered in the calculations.

IOA-000056764 (Repackager)

InstallShield can now extract an icon from a file when Repackager is used to repack an installation that contains an icon file that has an .exe file extension. Previously, build error -3204 occurred in this scenario, stating that the icon file could not be extracted.

IOA-000057596, IOA-000057630 (Basic MSI, InstallScript MSI, Merge Module, MSI Database, MSM Database, Transform)

If you use the custom InstallShield handling method for setting permissions for a registry key that uses a property in the key name, InstallShield can now correctly set the permissions. Previously, run-time error 27555 occurred.

When a component is installed to two different locations by two installations that have different product codes, and the component contains a file for which permissions are set using the custom InstallShield handling method, uninstalling one of the products no longer causes run-time error 27555.

IOA-000057977 (Basic MSI, Merge Module)

The Text setting has been removed from the edit field type of dialog control, since it is not used by the run time.

IOA-000058011 (Basic MSI, Transform)

The MSI Debugger no longer displays an "unknown source" error when debugging an installation that contains support files.

IOA-000058119 (InstallScript)

If you include a single pound sign (#) in the registry data value of a registry entry in the Registry view, the InstallScript installation no longer adds an extra pound sign (##) to the data at run time.

IOA-000058229 (Basic MSI, InstallScript, InstallScript MSI, Merge Module)

If you create a path variable that is relative to a base path variable (that is, you use a base path variable in the definition of the relative path variable), you can now change the value of the base path variable by passing the -l parameter to ISCmdBld.exe when building the release from the command line; that is, the definitions of the relative path variables are updated as expected. Previously in this scenario, the new value that was passed through the -l parameter was ignored, and the value that was set in the project file was used.

IOA-000058257 (InstallScript, InstallScript MSI)

If the InstallScript function DialogSetInfo is called with DLG_INFO_KUNITS, the installation no longer crashes when an end user specifies a drive in the SdFeatureDialog dialog.

IOA-000058457

If the InstallScript function FileInsertLine is called to insert a line in a text file, the file is now updated as expected. Previously in some cases, the file was re-created, but not all of the lines were written in the file. In addition, FileInsertLine no longer returns a success status when ListWriteToFile fails.

IOA-000058597 (Basic MSI)

Feature prerequisites are now installed during a multilanguage, silent installation that is launched from a Command Prompt window in which Setup.exe is not located in the working directory.

IOA-000058773 (Basic MSI, InstallScript MSI)

If you add a dynamically linked file to a component in the Setup Design view when the feature-component tree is expanded, InstallShield no longer collapses the feature-component tree.

IOA-000058952 (InstallScript Debugger)

If you add an InstallScript variable to the watch list in the InstallScript Debugger by right-clicking the selected variable in the script window and then clicking Add to Watch, the debugger no longer adds an extra space to the end of the variable name that it looks for. Previously, the debugger added the extra space, and this caused a "symbol not found" error.

IOA-000059066 (Basic MSI, InstallScript MSI, Merge Module)

If you are creating a managed-code custom action for a managed assembly that was created with .NET Framework 4, you can now browse to select the appropriate public method from the list of public classes that are available in the assembly. The browse functionality is available if the .NET Framework 4 is installed and if the location of the assembly is set to the Binary table or installed with the product. Previously, the browse functionality was not available in this scenario.

IOA-000059149 (Basic MSI, InstallScript MSI)

If a rollback is triggered by a failing custom action or the installation is cancelled by an end user, the IIS Web site can be rolled back now. Previously, in this scenario, if a Windows Installer property was used for the name of the Web site and the Web site's component contained child components, the Web site was not rolled back.

IOA-000059191 (Basic MSI, InstallScript MSI, Merge Module)

If you add to your project a managed-code custom action that uses the default method signature, the "Use custom method signature" check box is no longer selected on the Method Signature dialog box, which is launched when you click the ellipsis button (...) in the Method Signature setting of the custom action.

IOA-000059234 (Basic MSI, InstallScript, InstallScript MSI)

The InstallShield prerequisites that install Oracle 11g Instant Client and Crystal Reports require that you obtain redistributables from Oracle or Crystal Reports before you can use these prerequisites in your installations. The details pane that is displayed when you select these prerequisites in the Redistributables view or the Prerequisites view now explains this. Previously, the details pane did not specify this information. Note that you can hide or show the details pane by clicking the Show Details button in these views.

IOA-000059286 (Basic MSI, InstallScript MSI)

Managed-code custom actions no longer fail to run when they are scheduled during the UI sequence in a project that contains two or more SQL connections.

IOA-000059333 (Merge Module)

If you use the Files and Folders view to add to your Merge Module project a file in the [GlobalAssemblyCache] folder, InstallShield now shows the file that you added in this view. Previously, the file was not displayed in the GAC from within the Files and Folders view, but it was listed in the Components view.

IOA-000059371 (InstallScript)

The nvSize parameter of the InstallScript function RegDBGetKeyValueEx was changed to only return a value and not use it as input.

IOA-000059406 (InstallScript, InstallScript MSI)

Setup.exe no longer crashes with an unhandled exception error (0x80004005) on a machine that is running with display settings of 256 colors or fewer.

IOA-000059643 (Basic MSI, InstallScript MSI)

If a standard DLL custom action is scheduled after the built-in InstallShield custom action ISSQLServerCosting (which InstallShield adds to a project automatically when you use the SQL Scripts view), the standard DLL action no longer fails with an "unknown source" error.

IOA-000059668 (Basic MSI, InstallScript MSI)

If the value of a Windows Installer property contains a semicolon and the property is being set in a chained Windows Installer package, the semicolon is no longer changed to a space at run time.

IOA-000059765 (InstallScript)

If you create and apply multiple differential releases that contain differing subsets of features that were included in the base installation, error -5005 (0x80000ffff, a catastrophic failure error) no longer occurs when uninstalling the product.

IOA-000059799 (Basic MSI)

The Italian version of the CustomSetup dialog now displays the Italian translation of the word *Help* on the Help button instead of a question mark.

IOA-000060134 (InstallScript)

If an end user enters a SQL password that includes double quotation marks while running an installation whose project includes a SQL connection and script in the SQL Scripts view, run-time error 27502 no longer occurs.

IOA-000060427 (InstallScript, InstallScript MSI)

When the SdLogonUserInformation dialogs are added to a project and the SdLogonUserCreateUser dialog is edited, the dialog's OK button now works as expected at run time. Previously, clicking the OK button in the SdLogonUserCreateUser dialog at run time had no effect.

IOA-000060526 (Basic MSI, InstallScript, InstallScript MSI)

Microsoft SQL Server 2008 Express SP1 requires Windows Installer 4.5 and .NET Framework 3.5 SP1. Therefore, the InstallShield prerequisite (Microsoft SQL Server 2008 Express SP1 (x86 & x64Wow)) that installs this technology onto 32-bit and 64-bit (Wow) systems has been modified to have dependencies for Windows Installer 4.5 and .NET Framework 3.5 SP1.

IOA-000060782 (InstallScript, InstallScript MSI)

If you create and apply a full release that updates a product that was installed by an earlier InstallScript installation, the product name is now displayed correctly in the SdWelcome and status dialogs. The same thing is true for a minor upgrade that updates a product that was installed by an earlier InstallScript MSI installation. Previously, %P was displayed in these dialogs instead of the product name.

IOA-000061105 (Merge Module)

If you try to add a second predefined folder in the Files and Folders view of a Merge Module project, InstallShield now adds it to this view, and it no longer displays the following error: "The handle is invalid. MsiRecordGetString()." Previously in this scenario, if the "Clean up unused directories" check box on the Directory tab of the Options dialog box was selected when you tried to add a second predefined folder, InstallShield displayed the error and did not add the folder.

IOA-000061452

If you use the auto completion functionality in the InstallScript view to enter the InstallScript function IsObject in your script, InstallShield now uses the correct capitalization for this function. Previously, InstallShield used the wrong capitalization, and this caused a compile error.

IOA-000061459

If you use \N in the value of a string entry, it is now displayed as is at run time. If you use \n, it is replaced by a newline character at run time. Previously, both instances were replaced by a newline character.

IOA-000063073 (Basic MSI, InstallScript, InstallScript MSI)

The InstallShield prerequisites that install the Microsoft Visual C++ 2008 Service Pack 1 have been updated to include Microsoft's ATL security updates. The InstallShield prerequisites that install the Microsoft Visual C++ 2010, Visual C++ 2008 Service Pack 1, and Visual C++ 2005 Service Pack 1 redistributables have been updated to include Microsoft's MFC security updates.

IOB-000060489 (Basic MSI, InstallScript MSI)

If an application pool password was set in a base installation through a Windows Installer property, a minor upgrade no longer resets the password to garbled characters.

IOB-000060616 (Basic MSI)

If an installation uses a Windows Installer property for an IIS application pool and the password is stored in the registry, the password is now encrypted. Previously, the password was not encrypted.

IOC-000076708

The wording in the Run Commands area of the Preferences tab on the Options dialog box has been revised to clarify behavior. The state of the check box in this area determines whether InstallShield automatically uninstalls your product before you debug your Basic MSI or InstallScript MSI installation or before you rerun it by relaunching it from the Build menu.

The instructions in this area of the tab now state, "Specify what to do when debugging or when rerunning your setup from the Build menu." The check box under those instructions is now labeled, "Uninstall the product automatically before installing or debugging." Previously, the wording for the instructions and the check box label did not refer to debugging.

IOC-000086151 (Basic MSI, InstallScript MSI)

If a merge module that is included in an installation project as a dependency of another merge module, the dependency merge module's check box is now selected and disabled. Previously, the Microsoft C++ Runtime

Library 6.0 merge module was added to installation projects by the Crystal Report 8.5 Object Wizard, but the merge module was deselected in the Redistributables view; this caused InstallShield to crash.

IOC-000086189 (Basic MSI, InstallScript MSI)

The predefined system search for .NET Framework 3.5 SP1 searches for the presence of .NET Framework 3.5 SP1 on target systems. If it is not present, the installation displays a message stating that this version needs to be present for the installation to continue. Previously, the error message stated that .NET Framework 3.5 SP1 or greater needed to be present, but the condition checked for only the specific version.

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 [Q211162](#).