InstallShield 2015
Release Notes

originally released June 2015; updated to include SP1, released September 2015

Introduction

InstallShield is the industry standard for authoring high-quality Windows Installer– and InstallScript-based installations, as well as Microsoft App-V packages. InstallShield 2015 offers new features and enhancements that make it easy to use the latest technologies.

InstallShield 2015 includes support for Windows 10 and Visual Studio 2015.

For the latest information about InstallShield 2015, including updates to these release notes, see the online version of the InstallShield 2015 release notes.

Changes in SP1 (September 2015)

Support for Windows 10

InstallShield has support for Windows 10.

Support for Microsoft Visual Studio 2015

InstallShield includes support for Visual Studio 2015. You can create InstallShield projects from within this version of Visual Studio.

Support for Microsoft App-V 5.1

The Microsoft App-V Assistant in InstallShield includes support for creating virtual applications that can run on Microsoft App-V 5.1 clients.

New InstallShield Prerequisite for App-V 5.1

InstallShield includes a new InstallShield prerequisite that can be used when including a Setup.exe for a generated App-V 5.x package. Note that a Setup.exe setup launcher is required if the InstallShield prerequisite needs to be included in the release.

- Microsoft App-V 5.1 Desktop Client

The redistributable file for this InstallShield prerequisite is not available for download from within InstallShield, since you must obtain it from Microsoft. Once you obtain the redistributable from Microsoft, place it in the location that is displayed when you are editing the prerequisite in the InstallShield Prerequisite Editor. For more information on the prerequisites necessary, see https://technet.microsoft.com/en-us/library/mt346482.aspx.

Windows 10 Added to Operating System Options in App-V Assistant for App-V 5.x Versions

App-V 5.1 introduced support for Windows 10 32-bit and 64-bit operating systems. Accordingly, on the Package Information page of the Microsoft App-V Assistant, the list of available operating systems that are displayed when App-V 5.x is selected has been updated to include:

- Windows 10 32-bit
- Windows 10 64-bit

New InstallShield Prerequisites for Microsoft Visual C++ 2015 and .NET Framework 4.6

InstallShield includes new InstallShield prerequisites that you can add to Advanced UI, Basic MSI, InstallScript, InstallScript MSI, and Suite/Advanced UI projects:

- Microsoft Visual C++ 2015 Redistributable Package (x86)
- Microsoft Visual C++ 2015 Redistributable Package (x64)
- Microsoft .NET Framework 4.6 Full
- Microsoft .NET Framework 4.6 Web

These prerequisites install the appropriate technologies on supported target systems.

New Constant for the DialogSetInfo Function

A new constant is available for the nInfoType parameter of the InstallScript function **DialogSetInfo**:

DLG_INFO_ALTIMAGE_HIDPI—This constant specifies a high DPI image to be displayed in the dialog. High DPI image types supported include BMP, GIF, JPEG, PNG, and TIFF. If transparency is required, image types that support transparency (such as PNG) should be used and szInfoString should specify the name of the image to be displayed (optionally including the path) in the dialog. This parameter applies to all dialogs that display the standard installation image on the left side of the dialog.

When DLG_INFO_ALTIMAGE_HIDPI is passed in nInfoType, the following parameter values are expected:

- szInfoString—The name of an image file to display, optionally including the path. If no path to the file is specific, the file is assumed to be in SUPPORTDIR. If this file does not exist, DialogSetInfo returns ISERR_FILE_NOT_FOUND.
- nParameter—The DPI scaling percentage. For example, pass 200 for a 200% image scale, 150 for a 150% scale, etc. The minimum supported scaling value is 25. If 0 is passed for this value, no image is displayed. If DLG_INFO_ALTIMAGE_REVERT_IMAGE is passed, the previous image used is displayed.

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

Option Added to Disable Architecture Validation

The **None** option has been added to the **Architecture Validation** options in the **Releases** explorer to specify the bypassing of build-time architecture validation.

IOJ-1737342 (Advanced UI, Suite/Advanced UI)

When using release flags to exclude a feature from a release, builds now complete successfully without error.

IOJ-1737068

The Japanese version of the InstallShield Files View tab (in the Options dialog) has been fixed to display properly.

IOJ-1736989

The repair function of InstallShield 2014 repairs successfully when InstallShield 2014 and InstallShield 2015 are installed on the same machine.

IOJ-1736418

Adding Primary Output in the Files and Folders view will now correctly associate components to a feature. Building the project results in the Primary Output files being included and the component added for Primary Output is associated to a feature.

IOJ-1736192 (Advanced UI, Suite/Advanced UI)

The following is only a documentation change for SP1. The behavior as described is in InstallShield 2015.

InstallShield automatically handles an empty eligibility condition as a request to ensure that a later version of the package is not already present on target systems. You can augment this, for example with a Platform condition to prevent installation on unsupported platforms, without losing the automatic behavior.

For .exe packages, create an appropriate eligibility condition to ensure that the Advanced UI or Suite/Advanced UI installation does not downgrade your product by allowing an earlier version to install over a later version.

IOJ-1736095 (Basic MSI, InstallScript MSI)

InstallShield has been updated to allow for the cloning of PowerShell custom actions.

IOJ-1736093 (Basic MSI, InstallScript MSI)

If you add multiple PowerShell custom actions, each PowerShell custom action now runs for each of its unique scheduled PowerShell custom action.

IOJ-1735718 (Basic MSI, InstallScript MSI)

Registry permission changes under HKEY_CURRENT_USER now succeed even with Windows Update KB3072630 installed.

IOJ-1735499

Typically the target version in an OSD file is automatically determined during conversion to App-V 4.x package format. The version of the shortcut target file is used, or if the target file does not have a version, then a default value of '1.0' is used.

Now, a way to set a custom version by manually adding an entry to the ISVirtualShortcut table has been added.

For more information, see the "Advanced Table Settings for Conversion to Microsoft App-V" topic in the InstallShield Help Library.

IOJ-1735304

When creating a project and including the Microsoft ReportViewer 2012 prerequisite, the Microsoft ReportViewer 2012 prerequisite files are able to download successfully without any error.

IOJ-1734790

If you attempt to save a project with a corrupt .isproj file, the corrupt .isproj file is backed up as an .isproj.bak file in the same folder, and then replaced with a fresh .isproj file.

IOJ-1734359 (InstallScript)

Deleting a registry set from an InstallScript project successfully deletes the registry.

IOJ-1734069

If you add multiple DLL files or an .exe file to the "Destination computer's files" pane of the Files and Folders view and then modify the files so that the timestamp changes in Windows File Explorer, the timestamp is now accurately updated to reflect the most recent timestamp reflected in Windows File Explorer.

IOJ-1733919 (Basic MSI)

You can change the property in a SetProperty tied to a control event without error even if there are multiple events that uses the same property.

IOJ-1733677

InstallShield successfully opens the Files and Folders view when integrated with Visual Studio. Previously some projects would hang or crash when opening this view.

IOJ-1733107

InstallScript Language Support for Windows 10 documentation changes have been added to the InstallShield Help Library. The following sections have been updated: the "SYSINFO" section, the "FeatureFilterOS" section, and "ISOSL_WIN10" has been added to the "Predefined Constants" section.

IOJ-1733016 (Basic MSI, InstallScript MSI)

When building a project containing a .NET 2.0 assembly from MSBuild, build errors -6212 or -7325 no longer occur.

IOJ-1732915 (InstallScript)

The paths to source files are once again included in InstallScript build reports.

IOJ-1730272

For users of concurrent licenses, Server.ini now contains the proper server value after upgrading the InstallShield Standalone Build original release to the service pack release.

IOJ-1737768

The ReportViewer 2012 prerequisite in InstallShield 2015 has an updated dependency of: Microsoft SQL Server 2012 Express SP2 System CLR Types (x86).prq, which is now included in InstallShield. When you select ReportViewer 2012, the download will now succeed as a result of it successfully finding the dependent .prq.

IOJ-1737004 (Basic MSI, InstallScript MSI)

If you build a setup that includes a .NET Installer Class on a machine with Microsoft .NET 4.6, the Installer Class can now install correctly.

New Features in Original Release Version (June 2015)

Support for Windows 10–Based Systems

InstallShield has support for Windows 10.

Targeting Windows 10

On systems with Windows 10, the Windows Installer properties VersionNT and VersionNT64 indicate 603, which was originally introduced as the version number of Windows 8.1. Therefore, it is not possible to create conditions in an .msi package that specifically target Windows 10.

Since Windows Installer 5.0 and Windows 7, DLL custom actions in .msi packages are shimmed to block obtaining the operating system version; the APIs GetVersion, GetVersionEx, and RtlGetVersion return a Windows version of 6.0.6000, which was originally the version number of Windows Vista. Therefore, it is also not possible to obtain the actual version number of Windows from a DLL custom action or from an InstallScript custom action (which is implemented as a DLL).

Because of the aforementioned behavior in Windows Installer, it is not easily possible to detect what version of Windows on which an .msi package is running. In areas where you can specify target system OS requirements, such as the Installation Requirements page in the Project Assistant of Basic MSI and InstallScript MSI projects, the Windows 8.1 option has been renamed as **Windows 8.1 or Windows 10** to reflect the new run-time behavior.

InstallScript, Advanced UI, and Suite/Advanced UI projects provide the ability to properly detect which version of Windows (including Windows 10) is present on a target system. Thus, if your installation needs to target or exclude Windows 10, you may want to use these types of projects to create conditions under which .msi packages should be run based on the actual target system platform.

The InstallShield prerequisites that should be installable on Windows 10 have been updated so that they are installed on those systems if needed. Previously, the prerequisites may not have run by default on those systems.

InstallScript Language Support for Windows 10

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

- SYSINFO.WINNT.bWin10—This is a new SYSINFO structure member. If the operating system is Windows 10, this value is TRUE. (This is applicable to InstallScript event-driven code; it is not applicable to custom actions.)
- ISOSL_WIN10—This is a new predefined constant that is available for use with the FeatureFilterOS function and the SYSINFO structure variable. It indicates that the target system is running Windows 10.

Digital Signing Improvements

InstallShield includes several improvements for digitally signing your installations and files at build time.

Support for SHA-256 Digital Certificates

InstallShield now enables you to use digital certificates that use the SHA-256 hashing algorithm for signing your installations and files at build time.

SHA-256 is favored over SHA-1, which is being deprecated because of the potential for security vulnerabilities. Microsoft announced that Windows will stop trusting items that were signed and timestamped with SHA-1 certificates after January 1, 2016. In addition, certification authorities—the organizations that issue certificates—are phasing out the creation of SHA-1 certificates. Thus, it is recommended that you replace any SHA-1 certificates in your InstallShield projects with SHA-256 certificates. For the latest information and more specific details, check with your certification authority.

In InstallShield, to replace a SHA-1 certificate with a SHA-256 certificate for signing your releases, use the Signing tab in the Releases view to replace the reference to the current certificate with one for a SHA-256 certificate.

If your project is configured to sign with a SHA-256 certificate, InstallShield uses a SHA-256 hash in the signature of the files that it signs at build time. If your project is still configured to sign with a SHA-1 certificate, InstallShield uses a SHA-1 hash; in addition, use of a SHA-1 certificate now triggers build warning -7346 to alert you about the SHA-1 usage.

In earlier versions of InstallShield, InstallShield used a SHA-1 hash in the signature of files when signing with either SHA-1 and SHA-256 certificates.

Ability to Use a Certificate Store for Referencing Certificates

When you are specifying the digital signature information that you want to use for signing your files and installations, InstallShield now lets you reference a certificate store that contains the certificate that you want to use. This support is available as an alternative to specifying a .pfx certificate file on your machine.

To specify whether you want to use a certificate store or a .pfx certificate, use the Digital Certificate File setting on the Signing tab in the Releases view. When you click the ellipsis button (...) in this setting, a new Certificate Selection dialog box opens, enabling you to specify certificate information such as the store name (Personal, Trusted Root Certification Authorities, Enterprise Trust, Intermediate Certification Authorities), the store location (user or machine), and the subject that identifies the specific certificate that you want to use. As an alternative, you can specify on this dialog box the path and file name of a .pfx file that you want to use.

If you configure your project to use a certificate that was imported with password protection into a store, Windows prompts for the password at build time when InstallShield is attempting to sign your project's files. The strong key protection that Windows uses does not permit InstallShield to provide the password to the cryptographic provider.

Certificate store support is also available for signing patches and QuickPatch packages:

- To specify certificate store or .pfx certificate information for a patch, use the Digital Signing tab for a patch configuration in the Patch Design view.
- To specify certificate store or .pfx certificate information for a QuickPatch package, use the Build Settings area of the General Information view in the QuickPatch project. This area includes a Digital Signature tab that includes the new support.

In addition, the command-line tool iSign.exe, which is available for signing already-built InstallScript releases, has been updated to include support for using certificates in certificate stores.

Note that InstallShield no longer has support for signing with .spc and .pvk files.

Ability to Specify a Program Name for the UAC Dialog Box

The Signing tab in the Releases view includes a new Signature Description setting. Use this setting to specify the text that you want to be displayed to the right of the "Program Name:" label on the UAC dialog box for the Setup.exe files, the .msi file, and other installation files that InstallShield signs at build time. The UAC dialog box opens when an end user launches the signed file and elevated privileges are required.

If you leave the Signature Description setting blank, InstallShield uses the name of the file without its extension as the text on the UAC dialog box.

Ability to Digitally Sign Media Header Files with a .pfx File or a Certificate in a Certificate Store

InstallShield now has support for using .pfx files to sign media header files (.hdr files), which are used for the One-Click Install type of installation for InstallScript projects. As an alternative, you can use a certificate in a certificate store to sign media header files.

Previously, it was necessary to sign with .spc and .pvk files instead of .pfx files.

Automation Interface Support for Specifying Digital Certificate Information

The automation interface has support for specifying the .pfx file and the certificate store information. It also has support for specifying the signature description.

In Basic MSI, InstallScript, InstallScript MSI, InstallScript Object, and Merge Module projects, the ISWiReleases object includes two new read-write string properties. In Advanced UI and Suite/Advanced UI projects, the ISWiSuiteReleases object includes these same new properties.

- DigitalCertificateInfo—This property gets or sets either the path to the .pfx file on your machine, or to a string that indicates certificate store details.
- SignatureDescription—This property gets or sets the signature description.

This feature resolves the following issues: IOA-000066347, IOA-000080318, IOC-000089619, and IOJ-1700927.

Ability to View Both the 32- and 64-Bit Areas of the Source Machine's Registry on 64-Bit Development Systems

If you are using InstallShield on a 64-bit development system, the Registry view in InstallShield now displays both the 32-bit and 64-bit areas of your machine's registry:

- HKEY_LOCAL_MACHINE\Software
- HKEY_LOCAL_MACHINE\Software\Wow6432Node

This support makes it easier to develop installations on 64-bit machines, since it enables you to drag and drop entries from those source areas to the appropriate areas in the destination pane of this view.

Previously, if you were using InstallShield on a 64-bit development system, the source panes in the Registry view in InstallShield did not show any 64-bit data from the HKLM\Software part of the registry; in addition, the source panes displayed 32-data from the machine's HKLM\Software\Wow6432Node area in the HKLM\Software area.

Note that if you want your installation to install registry data to a 64-bit area of the registry on 64-bit target systems without having it redirected to a 32-bit area, you must place the registry data in a component that is marked as 64 bit. Simply dragging 64-bit data from the source panes in the Registry view to a location in one of the destination panes of the view does not mark the component as 64 bit.

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

Support for Embedding Formatted Expressions in Advanced UI and Suite/Advanced UI Projects for Run-Time Resolution

In various areas of Advanced UI and Suite/Advanced UI projects, you can now embed object expressions to query target systems for information about a file, a registry entry, the operating system, or other details. This enables you to dynamically configure many of the Advanced UI or Suite/Advanced UI settings at run time based on target system–specific conditions. Object expressions use this convention:

```
[@Object(Parameters, ...).Property(Parameters, ...)]
```

Each object expression contains a reference to an object, which is a collection of object-specific properties. Objects and properties may contain parameters.

For example, the following Platform object expression obtains the architecture (x86, x64, IA64, ARM, or Unknown) of the machine on which the Advanced UI or Suite/Advanced UI installation is running:

```
[@Platform.Architecture]
```

The following Registry object expression obtains the value data of the RegisteredOwner value for the HKLM\Software\My Company Name\My Product Name registry key:

```
[@Registry(HKLM\Software\[COMPANY]\[PRODUCT],
true).KeyValue(RegisteredOwner)]
```

You can embed object expressions within other formatted expressions, such as within property expressions or other object expressions. In the following expression, a Registry object expression is embedded as part of the parameter to a File object expression.

```
[@File([@Registry(HKLM\Software\MyProduct).KeyValue(MyProductPath)]\MyProduct.exe).Version]
```

If the MyProduct.exe file is in the location that is specified for the MyProductPath value data, the File object expression returns the version of the file. If the file is not found in that location, or if the registry value does not exist, the File object expression returns an empty string.

InstallShield also now enables you to pass literal square brackets as part of strings through the command line to packages in an Advanced UI or Suite/Advanced UI installation. For example, a command line such as **[\[]Text[\]]** resolves as **[Text]** at run time and is passed to the package with the square brackets. Previously, a string enclosed within square brackets was treated as a property that the installation attempted to resolve at run time. The only work around was to use a formatted property expression (such as **[PropertyForSquareBracketString]**) that would be resolved at run time to a property value that included the square brackets.

This feature resolves the following issues: IOA-000066232, IOA-000078276, IOA-000078602, and IOJ-1700875.

Ability to Share Common Packages Among Different Advanced UI and Suite/Advanced UI Installations

When you are configuring a package for an Advanced UI or Suite/Advanced UI installation, you can now mark it as shared. The shared package functionality helps to ensure that if two or more Advanced UI or Suite/Advanced UI installations are sharing a package, the package remains on the target system until all of the Advanced UI and Suite/Advanced UI products are removed.

To mark a package that is selected in the Packages view of an Advanced UI or Suite/Advanced UI project as shared, select Yes for the new Shared setting. In addition, ensure that the package GUID is the same across all Advanced UI and Suite/Advanced UI projects that are sharing the package; the Package GUID setting in the Packages view is where you can view and modify the GUID of a package.

Before this built-in shared support was available, unexpected results may have occurred in various scenarios. For example, in some scenarios, if two different Advanced UI or Suite/Advanced UI installations installed a shared package on a target system and then only one was uninstalled, the shared package was removed. The remaining Advanced UI or Suite/Advanced UI product may have been unusable because of the missing shared package. In other scenarios, the shared package was erroneously left behind on a target system, even though no Advanced UI or Suite/Advanced UI product remained.

Previously in some scenarios, before this built-in shared support was available, if two different Advanced UI or Suite/Advanced UI installations installed a shared package on a target system and then only one was uninstalled, the shared package was removed. The remaining Advanced UI or Suite/Advanced UI product may have been unusable because of the missing shared package.

This feature is available for the following types of packages in Advanced UI and Suite/Advanced UI projects: .msi, .msp, .exe, .appx, InstallScript, Basic MSI project, and InstallScript project.

This feature resolves the following issues: IOA-000080983 and IOJ-1666861.

Ability to Override the Product Name, Product Version, and Suite GUID for a Release in an Advanced UI or Suite/Advanced UI Project

InstallShield now lets you override the product name, product version, and Suite GUID values that are specified in the General Information view of your Advanced UI or Suite/Advanced UI project with a new value for a selected release. To override the General Information view values, use the new settings that are available on the Build tab for a release in the Releases view: Product Name, Product Version, and Suite GUID.

This feature resolves issue IOA-000077862.

Ability to Override the Values of Path Variables for a Release in an Advanced UI or Suite/Advanced UI Project

InstallShield now lets you override the values of your project's custom path variables (standard path variables, environment path variables, and registry path 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 want to use this functionality to swap out UI elements such as images and EULAs for different releases in your project; this would enable you to easily generate installations for different editions or branded versions of your product.

To override one or more path variables in your project, use the Path Variables Overrides setting, which is on the Build tab for a release in the Releases view.

This feature resolves issue IOJ-1659796.

Automation Interface Support for Advanced UI and Suite/Advanced UI Projects

InstallShield now enables you to automate most development and build processes of Advanced UI and Suite/Advanced UI projects (.issuite files) through the automation interface without having to directly open the InstallShield interface and make changes in different views. The automation interface provides programmatic access to various areas of Advanced UI and Suite/Advanced UI projects; it exposes a COM interface that you can call from many languages to create, edit, and build Advanced UI or Suite/Advanced UI projects.

The top-level automation object for Advanced UI and Suite/Advanced UI projects is the ISWiProject object. Sample VBScript code that opens, modifies, saves, and closes a project begins and ends like this:

```
Dim pProj As ISWiProject
Set pProj = CreateObject("ISWiAutoSuite22.ISWiProject")
pProj.OpenProject "C:\InstallShield 2015 Projects\Project1.issuite"
' perform changes here
pProj.SaveProject
pProj.CloseProject
```

By calling methods, getting and setting properties, and accessing collections, you can also add features and packages to your project, configure conditions, and more. For more information, see the "Automation Objects for Advanced UI and Suite/Advanced UI Projects" section of the InstallShield Help Library.

Note that the ProgID for the Advanced UI and Suite/Advanced UI automation interface is ISWiAuto**Suite**22.ISWiProject; the ProgID for other project types is simply ISWiAuto22.ISWiProject.

This feature resolves issue IOA-000077264.

New InstallShield Prerequisites for Microsoft Visual C++ 2013, .NET Framework 4.5.2, and More

InstallShield includes new InstallShield prerequisites that you can add to Advanced UI, Basic MSI, InstallScript, InstallScript MSI, and Suite/Advanced UI projects:

- Microsoft Visual C++ 2013 Redistributable Package (x86)
- Microsoft Visual C++ 2013 Redistributable Package (x64)
- Microsoft .NET Framework 4.5.2 Full
- Microsoft .NET Framework 4.5.2 Web
- Microsoft .NET Framework 4.5.1 Full
- Microsoft .NET Framework 4.5.1 Web
- Microsoft SQL Server 2012 Express SP2 (x86)
- Microsoft SQL Server 2012 Express SP2 (x86 & x64Wow)
- Microsoft SQL Server 2012 Express SP2 (x64)
- Microsoft SQL Server 2012 Express SP2 LocalDB (x86)
- Microsoft SQL Server 2012 Express SP2 LocalDB (x64)
- Microsoft SQL Server 2012 Express SP2 Management Objects (x86)
- Microsoft SQL Server 2012 Express SP2 Management Objects (x64)
- Microsoft SQL Server 2012 Express SP2 System CLR Types (x86)
- Microsoft SQL Server 2012 Express SP2 System CLR Types (x64)
- Microsoft SQL Server 2008 Express SP1
- Microsoft SQL Server 2005 Express SP3
- Internet Explorer 11.0 for Windows 7 (x86)
- Internet Explorer 11.0 for Windows 7 and Windows Server 2008 R2 (x64)
- Microsoft ReportViewer 2012

These prerequisites install the appropriate technologies on supported target systems.

The Microsoft SQL Server 2012 Express SP2 prerequisites replace the Microsoft SQL Server 2012 Express SP1 prerequisites.

This feature resolves the following issues: IOJ-1701054 and IOJ-1726208.

New Predefined System Searches for Internet Explorer 10 and 11

InstallShield has new predefined system searches that check target systems for Internet Explorer 10 or Internet Explorer 11. If your installation or product requires either of those versions, you can use the System Search view or the Installation Requirements page in the Project Assistant to add one of 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 change is available in Basic MSI and InstallScript MSI projects.

Support for Creating Virtual Applications for Microsoft App-V 5.0 SP3; Additional App-V Package Improvements

InstallShield and the Microsoft App-V Assistant in InstallShield include support for creating virtual applications that can run on Microsoft App-V 5.0 SP3 clients. In addition, the Microsoft App-V Assistant includes new settings and capabilities—some of which apply to App-V 5.0 SP3, and some of which apply to earlier versions of App-V.

New InstallShield Prerequisites for App-V 5.0 SP3

InstallShield includes new InstallShield prerequisites that you can add to Advanced UI, Basic MSI, InstallScript, InstallScript MSI, and Suite/Advanced UI projects:

- Microsoft App-V 5.0 SP3 Desktop Client (x86)
- Microsoft App-V 5.0 SP3 Desktop Client (x64)

The redistributable files for these InstallShield prerequisites are not available for download from within InstallShield, since you must obtain them from Microsoft. Once you obtain one of the redistributables from Microsoft, place it in the location that is displayed when you are editing the prerequisite in the InstallShield Prerequisite Editor.

Ability to Map Files into the Virtual File System Instead of a Primary Application Directory

The Microsoft App-V Assistant now enables you to configure your App-V package to map files into the virtual file system (VFS). This support is available for App-V 4.x and 5.x packages.

To specify whether you want to map the files to the VFS or use a primary application directory, use the Files page in the Microsoft App-V Assistant. The More Options area on this page has a new File Mapping link. When you click this new link, a new File Mapping dialog box opens, enabling you to select the appropriate option.

The File Mapping link and dialog box replace the Primary Application Directory link and dialog box, which previously enabled you to specify primary application directory.

Ability to Create App-V 5.x Packages that Have Full Write Permissions to the Virtual File System

The Microsoft App-V Assistant now enables you to specify whether you want an App-V 5.x package that you are creating to have full write permissions to the virtual file system (VFS). To specify whether you want to use this support, use the new **Allow full write permission to the VFS** check box. This check box is on the File Mapping dialog box, which is available from the Files page in the Microsoft App-V Assistant when you click the File Mapping link in the More Options area.

Ability to Configure Advanced COM Isolation Settings for App-V 5.x

The Microsoft App-V Assistant now enables you to configure advanced settings for COM isolation. This support is available for App-V 5.x packages.

To configure the new settings, use the Package Information page in the Microsoft App-V Assistant. The More Options area on this page has a new Isolation Settings link. When you click this new link, a new Isolation Settings dialog box opens. This dialog box enables you to specify whether you want to isolate the COM objects from the local system, or allow them to interact with the local system. This dialog box also lets you indicate whether you want to isolate named objects from the local system, or allow them to interact with the local system.

The Microsoft App-V Assistant is available in the Virtualization Pack.

This feature resolves the following issues: IOJ-1720473, IOJ-1724900, IOJ-1725148, and IOJ-1726018.

Enhancements

Performance Improvement for the Files and Folders View

InstallShield has been enhanced to more quickly load the Files and Folders view of large projects.

This enhancement resolves issue IOJ-1667312.

Ability to Filter Items in the Files and Folders View by Feature

The Files and Folders view now contains a View Filter that lists each of the features in your project, as well as an All Application Data option. You can use this filter to show and hide files and folders in the destination panes of this view:

- To see in this view only files and folders that belong to a particular feature, select that feature in the View Filter list.
- To add a file or folder to a specific feature, select that feature in the View Filter list. Then add the file or folder to the appropriate location in the Destination computer's folders pane.
- To see all of the files and folders that are in your project, select the All Application Data option in the View Filter list.

The new View Filter replaces the previous Add new components to the feature filter, which did not have support for showing and hiding files and folders in the view according to the selected feature.

This enhancement resolves issue IOA-000082226.

Ability to Access the Installer Session from PowerShell Scripts in Basic MSI and InstallScript MSI Installations

The PowerShell custom action support has been enhanced. It now includes support for several cmdlets that let you interact with the running Basic MSI or InstallScript MSI installation at run time. The cmdlets enable you to get and set Windows Installer properties, expand the value of formatted expressions, and write information to the log file.

With this revised implementation, you can use the Windows Installer property IS_CLR_VERSION to identify a semicolon-delimited list of .NET Framework versions that the custom action should attempt to load to run your PowerShell script.

This enhancement resolves issue IOA-000078121.

Ability to Configure Major Upgrades in Transform Projects and MSI Databases

The Upgrades view in Transform projects and in direct-edit mode for MSI databases now includes an Upgrades view that you can use to create major upgrades. To add an upgrade entry in a Transform project or in direct-edit mode for an MSI Database project, in the Upgrades view, right-click the Upgrade Windows Installer Setup node and then click Add Major Upgrade Item. Then configure its settings in the right as needed.

This enhancement resolves issue IOB-000062734.

Improvements to the ISHiddenProperties Property for Preventing Passwords from Being Written in Advanced UI and Suite/Advanced UI Log Files

The ISHiddenProperties property stores a semicolon-delimited list of case-sensitive property names whose values you do not want to be written to the debug log files. This property enables you to prevent properties that contain passwords and other sensitive information from being logged. The ISHiddenProperties property has been improved. You can now use this property for prevention of logging for values in the following scenarios:

- The end user sets the value of the property through command line when launching the Advanced UI or Suite/Advanced UI Setup.exe file.
- The property is configured through a command line that the Advanced UI or Suite/Advanced UI installation passes to a package. This is configurable in the Packages view, on the Common tab, in the Operation area.

Previously, ISHiddenProperties prevented logging only values that would be logged by changing the property value.

This enhancement resolves issue IOJ-1719756.

Improvements to the Interpretation of Ampersands in the Wizard Interface of Advanced UI and Suite/Advanced UI Installations

The interpretation of ampersands in certain areas of the wizard interface of Advanced UI and Suite/Advanced UI projects has been modified.

If you use an ampersand (&) in any of the following areas of the wizard interface of your installation, the installation now displays the ampersand as a literal character instead of interpreting it as a prefix character for a keyboard shortcut. The change also applies if any of these strings include a property that resolves to a value that contains an ampersand.

- The string in the header area of a wizard page or secondary window—This is configured in the Title setting for a wizard page or window in the Wizard Interface view.
- The caption bar of wizard pages—This is configured in the Wizard Caption setting for the Wizard Pages node in the Wizard Interface view.
- The supplemental explanation area of a command link control—This is configured in the Note setting for a wizard page or window.
- The alternate text for an image control—This is configured in the Alt Text setting of an image control on a wizard page or window.

For most label controls, True is now selected by default for the SS_NOPREFIX subsetting under the Style setting; previously False was selected by default. A True value for this subsetting ensures that any ampersands that are included in the string entries for such controls are not inadvertently interpreted as keyboard shortcuts, and that ampersands in the control's strings are displayed as ampersands in the wizard interface. This applies to the built-in default wizard pages and windows, as well as the predefined wizard pages. The only exceptions to this SS_NOPREFIX subsetting change are label controls that may contain keyboard shortcuts. For example, the default predefined Web Deploy wizard page has several text boxes with corresponding labels that may contain keyboard shortcuts. Therefore, False continues to be default value for the SS_NOPREFIX subsetting of such controls.

The InstallShield Help Library has a new "Designating Keyboard Shortcuts and Using Ampersands (&) in the Wizard Interface" help topic that explains how to configure your project to support the use of ampersands and keyboard shortcuts in your wizard interface.

This enhancement resolves issue IOJ-1722935.

Enhanced File-Related Types of Condition Checks Can Search Multiple Target System Paths in Advanced UI and Suite/Advanced UI Installations

When you are building a conditional statement for an exit, detection, eligibility, feature, or wizard interface condition in an Advanced UI or Suite/Advanced UI project, or for an action condition in a Suite/Advanced UI project, you can use the following file-related types of condition checks:

- File Exists—This type checks target systems for the presence of a particular file.
- File Comparison—This type checks target systems for specific information—date, version number, or component—for a particular file.

Both of these condition checks have been expanded to enable you to define conditions that check for the file in multiple paths on target systems. Previously, each condition could check only one specific location on target systems.

To enable this support, the existing Path setting for these condition checks has been expanded to let you enter only the name of the file (that is, optionally now without the path); if you leave out the path, you can use the new Search Path setting to specify a semicolon-delimited list of paths that you want the installation to check at run time for the specified file. You can include in this setting one or more formatted expressions that contain property names, environment variable references, and other special strings; at run time, the installation expands the values of these expressions.

For example, to search for the specified file in all directories that are defined for the PATH environment variable on a target system, as well as the path that is defined in the registry, enter the following value in the Search Path setting:

```
[%PATH];[@Registry(HKLM\SOFTWARE\MyPath).KeyValue(MyValue)]
```

Improvements for Importing InstallShield Prerequisites with Certain Types of File-Related Condition Checks in Advanced UI and Suite/Advanced UI Projects

Advanced UI and Suite/Advanced UI projects now have support for the following conditions that InstallShield prerequisites support:

- If you use the condition **A file does or does not exist** or **A file with a certain date exists** in a prerequisite and if you reference a registry key within square brackets for part of the path to the file, a Basic MSI, InstallScript, or InstallScript MSI installation resolves the registry key with the registry value. Now that Advanced UI and Suite/Advanced UI installations have expanded formatted-expression support for conditions, these types of prerequisite conditions are properly imported into Advanced UI and Suite/Advanced UI projects and resolved at run time. Previously, the registry key within square brackets was not resolved at run time; therefore, these types of conditions always evaluated as false.

- If you use the condition **A file does or does not exist** or **A file with a certain date exists** in a prerequisite and if you omit the path to the file, a Basic MSI, InstallScript, or InstallScript MSI installation searches for the specified file in all directories that are defined for the PATH environment variable on the target system. Now that Advanced UI and Suite/Advanced UI installations have expanded formatted-expression support for conditions, these types of prerequisite conditions are properly imported into Advanced UI and Suite/Advanced UI projects and resolved at run time. Previously, the path to the file could not be resolved; therefore, this type of condition always evaluated as false.

The InstallShield prerequisites for Oracle Instant Client are examples of prerequisites that use the file-related conditions. Now if you import these types of InstallShield prerequisites into Advanced UI or Suite/Advanced UI projects, those file-related conditions are configured properly in the Detection Condition setting of the Packages view.

Ability to Upgrade a Project Through the Automation Interface

The automation interface includes support for upgrading a project from an earlier version of InstallShield to the current version. To upgrade a project through the automation interface, use the ForceUpgrade method of the ISWiProject object.

This enhancement is available for the following project types: Advanced UI, Basic MSI, DIM, InstallScript, InstallScript MSI, InstallScript Object, Merge Module, and Suite/Advanced UI.

This enhancement resolves issue IOA-000124977.

Ability to Select the Virtual Machine Configuration for a Release Through the Automation Interface

The automation interface has support for selecting the virtual machine configuration for a given release; this enables you to specify which configuration you want to use for distributing the selected release to a virtual machine (VM) at build time.

In Basic MSI, InstallScript, and InstallScript MSI projects, the ISWiReleases object includes several new read-write properties. In Suite/Advanced UI projects, the ISWiSuiteRelease object includes the same new read-write properties.

- **VMConfig**—This property gets or sets the name of the group of VM configuration settings that you want to use when distributing a release to a VM.
- **VMSnapShot**—This property optionally gets or sets the name of the snapshot that you want to use for distributing the release.

  If no snapshot is specified for the VM configuration, InstallShield does not revert to a specific snapshot. InstallShield powers on the VM without reverting it to a specific snapshot, and copies your installation to the VM.
- **VMMachineUserName**—This property gets or sets the user name for the VM configuration.
- **VMMachinePassword**—This property gets or sets the password for the VM configuration.

- **VMStageMachineCopyPath**—This property gets or sets the location on the VM where the release is to be distributed. The last subfolder in the path can be a path that InstallShield creates on the VM; the other folders in the path must already exist.
- **VMStagePostBuild**—This Boolean property indicates whether you want InstallShield to automatically distribute the selected release each time that it is successfully built.

This enhancement is available in the Premier edition of InstallShield.

This enhancement resolves issue IOJ-1663785.

Enhancement to ISICE11

InstallShield internal consistency evaluator 11 (ISICE11) has been improved. If an executable file in your project includes a valid manifest that includes an asm.v2 entry for the trustInfo element, ISICE11 no longer occurs during validation. Previously, the asm.v3 entry was checked, but not the asm.v2 entry.

This enhancement resolves issue IOJ-1720356.

Enhanced Dialogs View in Basic MSI, DIM, and Merge Module Projects: All Behavior Settings for Each Control Are Shown in a Single Grid

The Dialogs view has been enhanced. All of the settings that are available for configuring the behavior (events, subscriptions, and conditions) of each user interface control on a run-time dialog are now displayed in a single grid. Previously, the settings were available through separate Events, Subscriptions, and Conditions tabs that were displayed in the bottom-right corner of the view.

To add an event, a subscription, or a condition to a control, in the Dialogs view, select the Behavior node under the dialog that contains the control. The controls that are used on the dialog are displayed in the center pane; select the control that you want to configure. Then use the Events, Subscriptions, and Conditions settings that are displayed in the right pane to configure the appropriate behavior.

This enhancement applies to the following project types: Basic MSI, DIM, Merge Module.

Enhanced Folder Views Show Total Count of Each Item in the Current Project

The folder views in InstallShield now enable you to see a quick summary of the contents of your project.

For example, the Application Data view—which is the folder view that contains the Files and Folders view and the Redistributables view in some types of projects—indicates the number of files, the number of merge modules, and the number of InstallShield prerequisites that are in the current project. The System Configuration view—which is the folder view that contains views such as Shortcuts and Registry— shows summary data such as the number of shortcuts, the number of registry keys, and the number of registry values.

This enhancement resolves issue IOJ-1724710.

New Machine-Wide Setting for Removing Unused Components Automatically

The Files View tab on the Options dialog box in InstallShield has a new **Clean up unused components** check box. This check box lets you specify whether you want InstallShield to remove unused components from your project automatically.

When this check box is selected, you delete all of the files in a component, and that component is not required elsewhere, that component is deleted automatically.

This new check box is a machine-wide setting. This check box is cleared by default; therefore, unused components are not removed automatically by default.

This enhancement resolves issue IOA-000081240.

Enhanced Behavior for Deleting Custom Actions that Use a File in the Binary Table

When you are deleting from your project a custom action that uses a file in the Binary table, InstallShield now determines whether the file in the Binary table is referenced by any other custom actions in the project. If it is not referenced by any other custom actions, InstallShield prompts you to specify whether you want to delete the entry from the Binary table while also deleting the custom action:

- The No button, which is the default selection, enables you to delete only the custom action.
- The Yes button enables you to delete both the custom action and the Binary table entry.
- The Cancel button enables you to avoid deleting the custom action and the Binary table entry.

If one or more other custom actions in the project reference the file in the Binary table, InstallShield deletes only the custom action that you chose to delete; it does not delete the Binary table entry. In this scenario, InstallShield does not display any prompt when you are deleting the custom action.

Previously, when you were deleting a custom action in the aforementioned scenario, InstallShield always displayed a prompt asking whether you wanted to delete the entry in the Binary table—regardless of whether any other custom actions in the project referenced it. The default button on this prompt was Yes, which resulted in both the custom action and the Binary table entry being deleted; this default behavior led to some users inadvertently deleting the Binary table entry.

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

This enhancement resolves issue IOB-000063404.

Improvement for Generating Concurrent Builds that Use the Same Custom Icon File for the Setup Launcher

If you generate builds concurrently, and the releases use the same custom icon file for Setup.exe, the builds complete successfully, and the setup launchers use the designated custom icon file. Previously in some cases, you may have encountered build warning -7212 ("The icon for the setup.exe could not be updated. Setup.exe will use the default icon.")

New Standalone Build Registry Entries Indicate InstallShield Version Information

The Standalone Build installation now installs the same InstallShield version–related registry values that the InstallShield installation installs under HKEY_LOCAL_MACHINE\SOFTWARE\InstallShield*VersionNumber*\Professional. This makes it possible to quickly verify whether all of the development and build machines in your environment are using the same versions of the product. The pertinent registry values are:

- Product Code—This registry value differs, depending on whether the InstallShield IDE or the Standalone Build is installed. It is updated for major upgrades of these tools.
- MPIndicator—If applicable, this registry value indicates the maintenance pack—for example, *Service Pack 1*.
- SP—If applicable, this registry value indicates the number of the service pack—for example, *1*.

For InstallShield 2015 and the InstallShield 2015 Standalone Build, the registry values are installed under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\InstallShield\22\Professional`
This enhancement resolves issue IOJ-1723003.

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

The following installations are available for download from the Flexera Software Product and License Center as documented in the InstallShield download and licensing instructions:

- InstallShield
- Redistributable files (for example, InstallShield prerequisites and InstallScript objects)
- Add-ons (if you are entitled to them) such as the Standalone Build, InstallShield Collaboration, and the InstallShield MSI Tools
- FlexNet Licensing Server software (if you purchased concurrent licenses and you need to set up your organization's license server)
- Skin Customization Kit
- InstallScript Object templates
- InstallShield service packs (if available)

Installing More than One Edition of InstallShield

Only one edition of InstallShield 2015—Premier, Professional, or Express—can be installed on a system at a time. In addition, the InstallShield 2015 DIM Editor cannot be installed on the same machine with any edition of InstallShield 2015.

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.

Installing More than One Version of InstallShield

InstallShield 2015 can coexist on the same machine with other versions of InstallShield.

The InstallShield 2015 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.

Project Upgrade Alerts

The following information describes possible upgrade issues that may occur when you upgrade projects that were created with InstallShield 2014 and earlier to InstallShield 2015. It also alerts you to possible changes in behavior that you may notice between new InstallShield 2015 projects and projects that are upgraded from InstallShield 2014 or earlier to InstallShield 2015.

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

If you use InstallShield 2015 to open an project that was created with an earlier version, InstallShield 2015 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 .775 (for an .ism project) or .2014 (for an .issuite project) before converting it. Delete the .775 or .2014 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 2015 projects in earlier versions of InstallShield.

You can upgrade projects that were created with the following versions of InstallShield to InstallShield 2015: InstallShield 2014 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 2015.

Changes to the List of Supported Versions of Windows for Target Systems

Windows XP SP3 and Windows Server 2003 SP2 are now the minimum versions of Windows that are required for target systems that run the installations that are created in InstallShield; this applies to all project types.

Changes to the List of Supported Versions of Visual Studio

Visual Studio 2010 is now the earliest version of Visual Studio that can be integrated with InstallShield Premier or Professional Editions; InstallShield no longer supports integration with Visual Studio 2008.

Removal of Support for Digitally Signing with .spc and .pvk Files

InstallShield no longer has support for using .spc and .pvk files to digitally sign files at build time.

If you configured a release or patch in InstallShield 2014 or earlier to be digitally signed at run time with .spc and .pvk files, and then you try to open that project in InstallShield 2015, InstallShield displays upgrade warning -6048. The warning explains that InstallShield is removing the .pvk file and the associated password from your project during the upgrade.

Before you can successfully build the release or the patch in InstallShield 2015, you will need to remove the .spc reference from the release or patch configuration. You can replace it with a .pfx certificate, or with a reference to a certificate in a certificate store.

If you try to build a release or a patch without removing the .spc reference from it, InstallShield displays build error -7347, stating that the .spc file must be removed.

To learn how to convert an .spc file and a .pvk file to a .pfx file, see "Digital Signing and Security" in the InstallShield Help Library.

Automation Interface Changes for Digital Signature Support

The ISWiRelease object no longer includes support for the following read-write properties:

- CorrespondingPrivateKey
- SoftwarePublishingCredentials

If you call those properties, you will encounter an error. Those properties have been replaced with the read-write string property DigitalCertificateInfo.

Removal of SignTool.exe and Signcode.exe from InstallShield Installation

SignTool.exe and Signcode.exe are no longer installed on your machine when you install InstallShield. If you want to digitally sign your files manually, consider using SignTool.exe, which is installed with Visual Studio and included in the Microsoft Windows Software Development Kit (SDK).

Changes to the PowerShell Support in Basic MSI and InstallScript MSI Installations

The PowerShell custom action support in Basic MSI and InstallScript MSI installations has been revised. The support no longer uses the Windows Installer property IS_PS_EXECUTIONPOLICY to indicate the name of the PowerShell execution policy that you want to be used to run PowerShell custom actions on target systems. Setting this property at run time no longer has any effect on the installation.

Changes to Default Eligibility Conditions for .msi Packages in Advanced UI and Suite/Advanced UI Projects

To make it possible to support shared packages in Advanced UI and Suite/Advanced UI projects, InstallShield no longer needs the previously required MSI Package eligibility conditions for .msi packages. Now you can limit the use of eligibility conditions of .msi packages to check for run-time environment requirements such as platform.

Now when you add an .msi package to an Advanced UI or Suite/Advanced UI project in InstallShield 2015, InstallShield no longer creates an MSI Package eligibility condition for that package by default. This behavior applies to new InstallShield 2015 projects, as well as projects that you created in InstallShield 2014 or earlier and then upgrade to InstallShield 2015.

In addition, if you upgrade an InstallShield 2014 or earlier project that contains the old default MSI Package eligibility condition for an .msi package to InstallShield 2015, InstallShield removes the unnecessary part of the condition from the Eligibility Condition setting for the package.

In some cases, if you customized the default MSI Package eligibility condition in InstallShield 2014 or earlier, InstallShield 2015 may remove it during the upgrade; in other cases, InstallShield leaves it as is. The actual upgrade behavior depends on the customization. For example, if you simply added a platform condition, InstallShield may be able to remove the original MSI Package part of the condition, leaving just the platform condition. However, if the customization is too complex, InstallShield leaves the condition as is.

It is recommended that you review the eligibility conditions for your .msi packages after upgrading to InstallShield 2015 to ensure that they are configured as expected.

Previously, the default MSI Package condition for the Eligibility Condition setting of .msi packages used asterisks (*) in the condition's Product Code and Product Version settings as placeholders for the package's own product code and product version.

Changes to the Interpretation of Ampersands in the Wizard Interface of Advanced UI and Suite/Advanced UI Installations

The interpretation of ampersands in certain areas of the wizard interface of Advanced UI and Suite/Advanced UI projects has been modified.

For new InstallShield 2015 projects, as well as projects that you created in InstallShield 2014 or earlier and then upgrade to InstallShield 2015, if you use an ampersand (&) in any of the following areas of the wizard interface of your installation, the installation now displays the ampersand as a literal character instead of interpreting it as a prefix character for a keyboard shortcut. The change also applies if any of these strings include a property that resolves to a value that contains an ampersand.

- The string in the header area of a wizard page or secondary window—This is configured in the Title setting for a wizard page or window in the Wizard Interface view.
- The caption bar of wizard pages—This is configured in the Wizard Caption setting for the Wizard Pages node in the Wizard Interface view.
- The supplemental explanation area of a command link control—This is configured in the Note setting for a wizard page or window.
- The alternate text for an image control—This is configured in the Alt Text setting of an image control on a wizard page or window.

Furthermore, for most label controls, True is now selected by default for the SS_NOPREFIX subsetting under the Style setting; previously False was selected by default. A True value for this subsetting ensures that any ampersands that are included in the string entries for such controls are not inadvertently interpreted as keyboard shortcuts, and that ampersands in the control's strings are displayed as ampersands in the wizard interface.

These SS_NOPREFIX changes apply to the built-in default and predefined wizard pages and windows that are available for new InstallShield 2015 projects. If you upgrade an InstallShield 2014 or earlier project to InstallShield 2015, the SS_NOPREFIX settings are changed automatically for the standard built-in default wizard page and windows. However, no changes are made automatically to custom wizard pages and windows—including the predefined wizard pages—when you upgrade these projects to InstallShield 2015; if you want to change the ampersand interpretation of these, you can do so manually.

Previously, to work around problems with an ampersand being used inadvertently to designate a keyboard shortcut in any of the aforementioned areas of the wizard interface, and instead show a single ampersand, it was possible to use an escaped ampersand (&&) in place of a single ampersand. However, if you upgrade a project that contains one or more instances of those workarounds to InstallShield 2015, the escaped ampersand is now displayed literally as two ampersands. Thus, it may be necessary to review the use of ampersands in the wizard interface of your projects and make changes as needed.

Changes to the Interpretation of Ampersands in the Setup Initialization and InstallShield Prerequisite Pending Dialogs for Basic MSI, InstallScript, and InstallScript MSI Installations

The interpretation of ampersands in certain areas of the run-time user interface of Basic MSI, InstallScript, and InstallScript MSI has been modified.

For new InstallShield 2015 projects, as well as projects that you created in InstallShield 2014 or earlier and then upgrade to InstallShield 2015, if you use an ampersand (&) in static controls of the initialization areas of the run-time UI of your installation, the installation now displays the ampersand as a literal character instead of interpreting it as a prefix character for a keyboard shortcut. The change also applies if any of these strings include a property that resolves to a value that contains an ampersand. Examples of areas that are affected are the text fields on the normal and small initialization dialogs, as well as the pending InstallShield prerequisite dialogs.

Previously, to work around problems with an ampersand being used inadvertently to designate a keyboard shortcut in the aforementioned areas of the UI, and instead show a single ampersand, it was possible to use an escaped ampersand (&&) in place of a single ampersand. However, if you upgrade a project that contains one or more instances of those workarounds to InstallShield 2015, the escaped ampersand is now displayed literally as two ampersands. Thus, it may be necessary to review the use of ampersands in the UI of your projects and make changes to string entries as needed.

Trialware Support

InstallShield no longer includes support for creating the Try and Die or the Try and Buy/Product Activation types of trialware. The Trialware view is no longer included in InstallShield.

Automation Interface Changes

If you use the automation interface with InstallShield or the Standalone Build, update your existing code to reflect the new ProgID: IswiAuto22.ISWiProject. The Standalone Automation Interface uses the same ISWiAutomation22.dll file that InstallShield uses, but it is installed to a different location.

Note that if you install the Standalone Build on the same machine as InstallShield, the last ISWiAutomation22.dll file that is registered is the one that is used.

Resolved Issues

IOA-000059322 (Advanced UI, Basic MSI, InstallScript MSI, Suite/Advanced UI)

If an installation caches a package in the SysWow64 folder on a 64-bit target system, the installation now runs it from the appropriate system folder. Previously on some 64-bit versions of Windows, the installation attempted to run the cached package from the System32 folder.

IOA-000064222 (Basic MSI, InstallScript MSI, Merge Module)

If you use the Data Languages setting on the Build tab in the Releases view to include in your release some components and exclude others based on each component's language, InstallShield now includes and excludes the appropriate components in the build output. Previously, InstallShield included all of the components, regardless of whether they were supposed to be excluded. To work around the issue, it was possible to use the Release Wizard to specify the languages of the components that needed to be excluded.

IOA-000065291

The StrLength help topic has been corrected. This help topic now states that the InstallScript function StrLength returns the number of code units before a null terminator in a Unicode (UTF-16) string. The help topic also has additional updated details on how to use this function. Previously, the help topic indicated that the function returns the number of bytes in a string.

The "Ensuring that Your InstallScript Code Supports Unicode" section in the "Upgrading Projects from InstallShield 2010 or Earlier" help topic has been expanded based on changes that were introduced in InstallShield 2011. It now explains that you may notice changes in the return values for StrLength between InstallShield 2010 and earlier projects and InstallShield 2011 and later projects.

IOA-000066977

The sample code in the "UseDLL Example" and "UnUseDLL Example" help topics has been revised. The erroneous semicolon in the #define lines of code has been deleted. In addition, the references to a MyDLL.dll file have been changed to a different file name, since the code is not intended to be used with the MyDLL.dll file that installed in the Samples folder that ships with InstallShield.

IOA-000073666

The description of the /uninst command-line parameter in the "Setup.exe and Update.exe Command-Line Parameters" help topic has been expanded. The description now explains that this parameter is available for compatibility with earlier versions of InstallShield. In addition, it states that in InstallScript MSI installations, the /uninst parameter does not invoke the InstallScript engine uninstall behavior that processes install log and removes any resources that were created and logged through script; the /removeonly option invokes this behavior.

IOA-000073755 (Basic MSI, InstallScript, InstallScript MSI)

If you pass the /runfromtemp command-line parameter to a Setup.exe file on a medium such as a DVD or a CD, and if the installation includes an InstallShield prerequisite that is configured to be copied from the source media, the prerequisite installation no longer fails with an error such as, "The files for installation requirement *MyPrerequisite* could not be found."

IOA-000074858

The description of the listAdd parameter in the ListAddList help topic has been revised to clarify the behavior of this parameter of the InstallScript function ListAddList. The description now states that this parameter contains the elements to be added to the destination list. ListAddList adds all elements (from the current element in this list to the last element in this list) to the destination list that the ListDest parameter identifies.

IOA-000075571 (InstallScript)

If Yes is selected for the 64-Bit Component and Self-Register settings for a component that is configured to install a COM server in WINSYSDIR64 (the 64-bit System32 folder), the COM server is now successfully registered. Previously, self-registration failed for COM servers with the WINSYSDIR64 destination.

IOA-000075693

The Reg-Free COM Wizard now creates a valid manifest file without displaying the error "An unexpected error encountered while attempting to create Reg-Free COM manifest file." Previously the wizard could not create a manifest file in some scenarios.

IOA-000076709 (Advanced UI, Suite/Advanced UI)

If an .exe package in an Advanced UI or Suite/Advanced UI installation is running while the progress wizard page is displaying billboards, the wizard UI no longer locks up until the .exe package exits. In addition, the wizard UI now switches to the next billboard.

IOA-000076793 (Advanced UI, Suite/Advanced UI)

If you use a .png file that has a transparent background as the resource for an image button control on a wizard page or secondary window, the transparent areas are now displayed with transparency. Previously, the transparent areas were displayed as white areas.

IOA-000081819 (InstallScript MSI)

Multilanguage InstallScript MSI installations that are compressed no longer exit unexpectedly after the PreparingToInstall dialog is displayed. Previously, the installations failed. This applies to InstallScript MSI projects in which the InstallScript UI style is the new style (which uses the InstallScript engine as an embedded UI handler).

IOA-000082218 (Advanced UI, Suite/Advanced UI)

When an Advanced UI or Suite/Advanced UI installation includes support for more than two languages, the language list on the InstallationLanguage wizard page no longer shows empty list items or extra space at the bottom of the drop-down list.

IOA-000082572 (Transform)

If you create a transform project that contains 150 MB to 2 GB worth of files that are compressed into a single .cab file, InstallShield creates the .cab file successfully. Previously, the .cab file was not created successfully and error 1334 occurred at run time, stating that the a file could not be found in the .cab file.

IOA-000082927 (Basic MSI, InstallScript MSI)

If you use the -t command-line parameter to specify the 32-bit location of the .NET Framework utilities Regasm.exe and InstallUtilLib.dll, InstallShield now uses the 64-bit versions of these utilities at build time for releases that include .NET installer classes and COM interop in 64-bit components. Previously, build error -6209 occurred, indicating that Regasm.exe could not be found.

IOA-000083152 (InstallScript)

If the value of the .NET Assembly setting of a component in an InstallScript project is Local Assembly, and if the installation is run on a target system that has .NET Framework 4.0 or later, the installation no longer triggers a prompt for .NET Framework 3.5.

IOA-000084393 (Advanced UI, Suite/Advanced UI)

If a command line for an InstallShield prerequisite includes an environment variable reference such as %temp%, and if you import that prerequisite into an Advanced UI or Suite/Advanced UI project, InstallShield now imports the command lines for that prerequisite with the correct syntax. This ensures that the appropriate command line is passed to the prerequisite package at run time. InstallShield now uses syntax such as [%temp] for the package in the in the Advanced UI or Suite/Advanced UI project.

Previously, after importing such a prerequisite into an Advanced UI or Suite/Advanced UI project, it was necessary to manually edit the command lines for the package so that it included a property in square brackets (such as [TempFolder]) instead of a reference to an environment variable.

IOA-000124593 (InstallScript MSI)

If your InstallScript MSI installation includes billboards and uses the STATUSBBRD progress dialog, the billboards are now displayed for the entire time that files are being transferred to target systems. Previously, partway during file transfer, the installation started displaying a blank area instead of the billboards on the progress dialog.

IOC-000078542

The InstallShield Help Library contains a new help topic called "Special Considerations for Searching the Registry at Run Time" that explains how Windows Installer stores registry entry types in a system search's property.

IOC-000090555

The "XCopyFile" help topic in the InstallShield Help Library has been expanded with details about the ability to specify a valid URL for the szSrcFile parameter.

IOC-000090932 (Advanced UI, Suite/Advanced UI)

If a file in the Interm folder of the project is read-only, the build now fails with a -7348 build error; the error message indicates which file could not be saved. Previously, the build failed with error code 0 and no error message.

IOJ-1660097 (Basic MSI, DIM, InstallScript MSI, Merge Module, Transform)

If you delete a component that contains a text file change that was configured in the Text File Changes view, InstallShield now removes the corresponding ISSearchReplace and ISSearchReplaceSet entries from your project.

Previously, InstallShield left the ISSearchReplace entry in the project and associated it with a nonexistent text change replacement set called NoISSearchReplaceSet. Therefore, before it was possible to configure a new text file change with the same text change name, it was necessary to delete the ISSearchReplace entry from the project through the Direct Editor view.

IOJ-1666040

If you try to install the InstallShield IDE on Windows XP or Windows Server 2003, you will now encounter an error stating that you need Windows Vista or later on your machine.

IOJ-1660222

The path for the VMConfigurations.xml file that is installed with InstallShield has been corrected in the "Sharing Virtual Machine Setting for Distribution of Releases" help topic; the file is in the InstallShield *Program Files Folder*\Support\0409 directory or the *Program Files Folder*\Support\0411 directory.

IOJ-1660957 (Basic MSI, InstallScript MSI)

The entire value of the Japanese version of the IDS_PREVENT_DOWNGRADE_EXIT string entry is now displayed at run time. Previously, part of the string entry's value was cut off.

IOJ-1661825

When you are using the Japanese ISCmdBld.exe to build from the command line, the Command Prompt window and the build log file now display Japanese characters instead of question marks.

IOJ-1662504

A corrected translation of the string "Welcome to the InstallShield Wizard for [ProductName]" has been incorporated into the default Italian run-time strings. Previously, the translation was the Italian equivalent of "InstallShield Wizard for [ProductName]."

IOJ-1663112

If you override the value of a path variable through the .isproj file for MSBuild, MSBuild now uses the new path variable value in other path variables that reference the one you have overridden. Previously, builds through MSBuild failed in some cases. For example, if the Link To setting for a file in your project was a path variable whose value referenced a path variable that you had overridden, the build previously failed.

IOJ-1664029 (Basic MSI, InstallScript MSI)

If you pass /? to Setup.exe when launching it from the command line, the help text that is displayed no longer includes a period after the silent mode sample: /s /v/qn

IOJ-1664860

The "Shipping Redistributable Files" help topic in the InstallShield Help Library now includes the SetupPrereq.exe file in the list of redistributable files that you can use according to the InstallShield End-User License Agreement.

IOJ-1665546

The Standalone Build now suppresses build errors about outdated merge module dependencies that are not found at build time. This error suppression behavior matches that of builds in the full version of InstallShield.

IOJ-1666524 (Suite/Advanced UI)

Suite/Advanced UI installations now handle positive exit codes from .exe actions correctly.

IOJ-1667107 (Transform)

If you add multiple transforms to your project through the Additional Transforms panel in the Open Transform Wizard, all of them are now displayed on this panel in the wizard. Previously, a display problem prevented more than one transform being listed on this panel.

IOJ-1667245

In the Direct Editor view, the horizontal scroll bar is no longer missing from certain tables.

IOJ-1667248 (Basic MSI, InstallScript MSI)

When an InstallScript custom action calls the InstallScript function CopyFile to attempt to overwrite an existing locked file on a target system, the function now returns an error code. Previously, the function did not return, which caused the installation to stop responding.

IOJ-1699692

The SelectDir and SelectDirEx help topics in the InstallShield Help Library now state that Windows displays the SelectDir and SelectDirEx dialogs; therefore, the installation cannot change the text of the buttons on these dialogs. Windows displays the button text in the language of the operating system.

IOJ-1700707 (Advanced UI, Suite/Advanced UI)

If the wizard interface includes a check box control that has a value of True for its BS_MULTILINE style setting and if the Text Style setting for the control is set, the text for the control wraps as needed to multiple lines. Previously, the text did not wrap to additional lines beyond the first.

IOJ-1700869 (Basic MSI, DIM, InstallScript, InstallScript MSI, InstallScript Object, Merge Module)

If a dependency option is selected in the .NET Scan at Build setting for a component that depends on a .NET 4.0 assembly in the GAC, InstallShield is now able to locate the assembly at build time. Previously, InstallShield was unable to find the assembly, and build warning -6248, stating that it could not find the dependent file or one of its dependencies.

IOJ-1718441 (Basic MSI, DIM, InstallScript MSI, Merge Module, Transform)

If you configure permissions for one folder in a project and then cancel out of configuring permissions for a second folder, the permissions for the first folder are no longer erroneously removed from the project. Previously, the permissions were removed in some circumstances.

IOJ-1719333 (InstallScript)

If you use the XML File Changes view to configure run-time changes to XML files, the installation and the uninstallation no longer leave behind related temporary support files in the %TEMP% folder.

IOJ-1721109 (Advanced UI, Basic MSI, InstallScript, InstallScript MSI, Suite/Advanced UI)

The conditions that are configured for the following InstallShield prerequisites have been corrected:

- Microsoft Visual C++ 2010 Redistributable Package (x86)
- Microsoft Visual C++ 2010 Redistributable Package (x64)

Windows 8 and Windows Server 2012 and later do not support these packages. Therefore, the prerequisites no longer target these platforms.

IOJ-1721241 (Basic MSI)

Error codes are no longer missing from localized error messages that are displayed in the user interface at run time for some run-time languages. Previously, the run-time strings included double curly braces ({{}}) around the error code. Text within double curly braces is visible only in the log file; it is not displayed in the UI.

IOJ-1722103 (InstallScript MSI)

The ISSelfRegisterCosting custom action no longer fails for a particular executable file when the value in the CmdLine column of the ISSelfReg table is set to the following:

```
/ServiceNoControlManager|/UnregServerX
```

IOJ-1722201 (InstallScript)

If the following conditions are true, fonts are properly installed to the Windows Fonts folder as expected:

- The InstallScript installation contains two features that contain the same component that is configured to install fonts to the Windows Fonts folder. The second feature is selected to be installed, but the first feature is not selected.
- Microsoft security update KB2993651 has been applied to the target system

Previously in this scenario, the fonts would not install, and a file error ("The system cannot find the file specified. (0x2)") was displayed.

IOJ-1722765 (InstallScript)

The entire status dialog is now displayed centered on the screen when the installation calls the InstallScript functions PlaceWindow and SizeWindow with the MMEDIA_AVI constant. Previously on Windows 7–based systems, part of the status dialog was cut off, and it was not displayed in the center of the screen.

IOJ-1723383, IOJ-1725814 (InstallScript Object)

InstallShield now builds InstallScript Object projects successfully. Previously, InstallShield crashed while building this type of project.

IOJ-1723476

In the InstallShield Prerequisite Editor, if you modify the path of a file on the Files to Include tab, the Prerequisite Editor no longer erroneously adds an extra backslash after a path variable.

IOJ-1723693 (InstallScript MSI)

If your InstallScript MSI installation uses InstallScript code to create keys in the 64-bit area of the registry during a major upgrade on a target system where the registry keys did not exist before the base installation was run, the major upgrade no longer displays a "Setup Launcher Unicode has stopped working" error message.

IOJ-1723921 (InstallScript)

Deleting a parent folder from a project now also deletes its subfolders. Previously, the subfolders remained in the project, and these subfolders were installed on target systems at run time.

IOJ-1724095 (Advanced UI, Suite/Advanced UI)

Build error -7239 no longer occurs if the file name of a package in the project contains more than 63 characters.

IOJ-1724624 (Suite/Advanced UI)

When a Suite/Advanced UI project includes an InstallShield project (.ism) as a package, the command-line build tool ISCmdBld.exe can now successfully build a release for this project. Previously, the build stopped abruptly, and it did not generate any output.

IOJ-1725037

The following InstallShield prerequisites now run on Windows 7– target systems.

- Microsoft SQL Server 2012 Express SP1 (x64)
- Microsoft SQL Server 2012 Express SP1 LocalDB (x64)
- Microsoft SQL Server 2012 Express SP1 Management Objects (x64)
- Microsoft SQL Server 2012 Express SP1 System CLR Types (x64)

Previously, these InstallShield prerequisites were configured to require Windows 7 SP1 or later.

IOJ-1725162

The "Setup.exe Return Values and Run-Time Errors (Advanced UI and Suite/Advanced UI Projects)" in the InstallShield Help Library contains troubleshooting tips on a number of previously undocumented exit and error codes.

IOJ-1726094

In the Professional edition of InstallShield, the About InstallShield dialog box (which is displayed when you click About InstallShield on the Help menu) no longer erroneously includes the string "with Virtualization Pack."

IOJ-1726115

Some index and search issues in the InstallShield Help Library have been resolved. The Index tab is no longer missing entries for the Windows Installer help. In addition, the Search tab once again displays search results for terms such as "ISWiProject Object."

IOJ-1726120 (InstallScript, InstallScript MSI)

In installations that use the blue dialog skin, the left side of the status dialog is no longer missing an image.

IOJ-1726328

The built-in Portuguese run-time language strings no longer contain extra quotation marks.

IOJ-1727345 (Suite/Advanced UI)

When a managed-code custom action in a Suite/Advanced UI installation returns a non-zero value, the installation now fails; previously, the installation ignored such return values.

IOJ-1727444 (InstallScript, InstallScript MSI)

Check boxes on run-time dialogs no longer appear disabled and grayed out on Windows 7–based target systems.

IOJ-1727555 (InstallScript)

The InstallScript system variable PROGRAMFILES64 is now resolved to the correct locations at run time on 64-bit target systems. Previously, if a base installation with the system variable was created in InstallShield 2014 or earlier, and a newer version was created in InstallShield 2014 SP1, PROGRAMFILES64 resolved to a 32-bit location during the update.

IOJ-1728369 (Basic MSI, InstallScript, InstallScript MSI, Suite/Advanced UI)

If you specify an invalid destination path as the location on a virtual machine where you want the release to be distributed at build time, InstallShield no longer becomes unresponsive.

IOJ-1728459 (Basic MSI, InstallScript, InstallScript MSI, Suite/Advanced UI)

When a release is configured to be distributed to a VMware Workstation at build time, the build log no longer contains the following error:

```
GetImages FindItems VIX_FIND_REGISTERED_VMS VIX error 6
```

IOJ-1728706 (Basic MSI, InstallScript, InstallScript MSI, Suite/Advanced UI)

If you enter credentials in the User Name and Password subsettings that are under the Configuration setting in the Virtual Machine area (on the Events tab for a release in the Releases view) instead of in the Edit Virtual Machine Configurations dialog box, InstallShield now uses those credentials when connecting to the virtual machine to distribute the release.

IOJ-1729373 (Basic MSI, InstallScript, InstallScript MSI)

If you use an ampersand (&) in your product name, the ampersand is now included as part of the product name in the small initialization dialog. Previously, it was omitted.

IOJ-1729395 (Basic MSI, DIM, InstallScript, InstallScript MSI)

The SQL script error handling support no longer ignores errors that are encountered in the CATCH block of TRY-CATCH statements at run time.

IOJ-1729415 (Basic MSI, DIM, InstallScript MSI)

If a SQL script that is included in the SQL Scripts view of your project contains more than one PRINT statement, all of the PRINT statements are now written to the log file now. Previously, only the first PRINT statement was logged.

IOJ-1729563 (InstallScript MSI)

The Korean translations for the IDS_ERROR_27503 and IDS_SQLLOGIN_TITLE string entries have been corrected.

IOJ-1729622 (Advanced UI, Suite/Advanced UI)

Advanced UI and Suite/Advanced UI projects now properly bind to source control in Visual Studio solutions. Previously, the binding support was not available; each time that the project is checked in to source control, a source control error was displayed.

IOJ-1730277 (Advanced UI, Suite/Advanced UI)

If you had manually modified the XML of a project file (.issuite) so that a package in the project to associate a feature in the package with a feature in the Advanced UI or Suite/Advanced UI project, and then you upgrade that project, InstallShield will no longer crash at build time.

IOJ-1730288 (InstallScript)

An abort during the OnFirstUIAfter event of an InstallScript installation that includes the Merge Module Holder Object no longer causes Setup.exe to crash.

IOJ-1730620 (Advanced UI, Suite/Advanced UI)

If you create one version of an Advanced UI, Suite/Advanced UI project and then create an upgrade with different values for the Disable Change Button setting or the Disable Remove Button setting in the Add or Remove Programs area of the General Information view, the installation now updates the product's Add or Remove Programs entry as configured in the upgrade.

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 Vista
Windows Server 2008
Windows 7
Windows Server 2008 R2
Windows 8
Windows Server 2012
Windows 8.1
Windows Server 2012 R2
Windows 10

Privileges

Administrative privileges on the system

Mouse

Microsoft IntelliMouse or other compatible pointing device

Optional Integration with Visual Studio

The following versions of Microsoft Visual Studio can be integrated with InstallShield Premier or Professional Editions:

Visual Studio 2010
Visual Studio 2012
Visual Studio 2013
Visual Studio 2015

The following editions of these versions of Visual Studio can be integrated with InstallShield Premier or Professional Editions:

Professional
Premium
Ultimate

For Target Systems

Target systems must meet the following minimum operating system requirement:
Windows XP SP3
Windows Server 2003 SP2
Windows Vista
Windows Server 2008
Windows 7
Windows Server 2008 R2
Windows 8
Windows Server 2012
Windows 8.1

Windows Server 2012 R2
Windows 10

Target systems must also support the SSE2 instruction set.

Known Issues

For a list of known issues, see Knowledge Base article 000017928.

Legal Information

Copyright Notice

Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see http://www.flexerasoftware.com/intellectual-property. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.