

InstallShield 2014

Release Notes

originally released May 2014; updated to include SP1, released December 2014

Introduction

InstallShield is the industry standard for authoring high-quality Windows Installer- and InstallScript-based installations, as well as Microsoft App-V packages. InstallShield 2014 offers new features and enhancements that make it easy to use the latest technologies.

For the latest information about InstallShield 2014, including updates to these release notes, see the [online version of the InstallShield 2014 release notes](#).

Changes in SP1 (December 2014)

You can obtain the installations for InstallShield 2014 SP1 through the [Flexera Software Product and License Center](#). For instructions, see the [download and licensing instructions](#) for InstallShield.

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.

Updates to Portuguese Run-Time Strings

The default run-time strings for Portuguese (Brazil) and Portuguese (Portugal) have been updated to reflect the Portuguese Language Orthographic Agreement of 1990 that finishes going into effect at the end of 2014, per a 6-year transition period starting in 2009.

This change resolves issue IOJ-1719546.

IOA-000079064, IOA-000079310 (Basic MSI, InstallScript MSI)

InstallShield now enables you to select several different versions of Windows on the Installation Requirements page of the Project Assistant. Previously, if you attempted to select a third check box, InstallShield displayed an error ("Field 'Condition' of table 'LaunchCondition': The string is too long for the length specified by the column definition") and would not enable a third check box to be selected.

IOJ-1660031 (Basic MSI, InstallScript, InstallScript MSI)

If you are using the Japanese version of InstallShield from within Visual Studio and you try to open the InstallScript view, InstallShield no longer displays an "unspecified error," preventing any changes to the InstallScript code.

IOJ-1661158 (Basic MSI, InstallScript, InstallScript MSI, Suite/Advanced UI)

InstallShield Premier Edition no longer encounters error 28 (out of stack space) when distributing a release at build time to a VMware ESX or ESXi Server.

IOJ-1662589, IOJ-1664180, IOJ-1664871, IOJ-1667078 (Basic MSI, DIM, InstallScript, InstallScript MSI, Merge Module)

InstallShield no longer crashes and .NET dependency scanning no longer fails at build time when certain files are included in the project or in a merge module that is part of the project.

IOJ-1662934 (Basic MSI, DIM, InstallScript MSI, Merge Module, Transform)

If you use environment variable formatting (for example, %MyEnvironmentVariable%) for a setting in the Text File Changes view, the name of the environment variable is now replaced with its value at run time when the installation is modifying the text file. Previously, the environment variable was not resolved.

IOJ-1663062 (InstallScript MSI)

If you create an InstallScript MSI installation in which the InstallScript UI style is the traditional style (which uses the InstallScript engine as an external UI handler) and it is run as a major upgrade, the upgrade now occurs as expected; it no longer crashes at the beginning of the Install sequence. Previously, the upgrade crashed, and a "Setup Launcher Unicode has stopped working" error was displayed.

IOJ-1663885 (Basic MSI, InstallScript MSI)

If you configure your patch to be signed with a digital certificate that uses a password, InstallShield now signs your patch at patch creation time. Previously, the patch was not signed.

IOJ-1664081 (InstallScript)

The InstallScript system variables `FOLDER_APPLICATIONS64` and `PROGRAMFILES64` are now resolved to the correct locations at run time on 64-bit target systems.

IOJ-1664236

If you build a Visual Studio solution that contains an InstallShield project, the appropriate release (the one that is associated with the active solution configuration) in the InstallShield project is now built. Previously, the release that was selected in the Releases view was built, regardless of which release was associated with the active solution configuration.

IOJ-1665093 (Advanced UI, Suite/Advanced UI)

If you create an Advanced UI or Suite/Advanced UI installation and configure it to include support for downloadable updates, the updates no longer fall back to the downloadable setup launcher file that was identified in the Update URL setting when the `isupdate.xml` file is found. Previously, if none of the updates that were listed in `isupdate.xml` were applicable, the Suite engine would try to download the `.exe` that was identified in the Update URL setting and run it anyway. In certain circumstances, this resulted in running an update that `isupdate.xml` identified as an older version.

IOJ-1665102 (InstallScript, InstallScript MSI)

When an installation is run on a Windows 7–based machine that is using the Windows Classic display theme, the check boxes are now displayed on the `SdFeatureDialog`.

IOJ-1665767 (Basic MSI, DIM, InstallScript, InstallScript MSI, Merge Module, Transform)

If you use the XML File Changes view to specify changes to an XML file and you indicate that you do not want the XML file to be formatted after the run-time changes are made, the file is no longer formatted at uninstallation time.

IOJ-1680657 (Suite/Advanced UI)

When an InstallScript action in a Suite/Advanced UI installation calls `CopyFile` or another InstallScript function that is valid in Suite/Advanced UI installations, the InstallScript action no longer tries to call into the user interface. Previously, the call to the UI failed because the UI is not available to

InstallScript actions in Suite/Advanced UI installations; as a result, the installation stopped responding.

IOJ-1699515 (Advanced UI, Suite/Advanced UI)

If an end user clicks the Print button on the LicenseAgreement wizard page and then selects a printer whose name contains a space, the selected printer prints the EULA without error. Previously, an error occurred, and the default printer printed the EULA.

IOJ-1700063 (Basic MSI)

If multiple instances (each sharing the same upgrade code) of a product are installed, the base one is uninstalled, and then the end user launches the setup launcher again, the instance selection dialog that opens no longer shows a duplicate entry of an instance under the **Maintain or upgrade an existing instance** option.

IOJ-1700373 (Basic MSI, DIM, InstallScript MSI, Merge Module, and Transform)

When a shortcut that uses an icon file (.ico) with different resolutions of images is installed and the end user views the shortcut in an Explorer window that uses a large icon layout, the highest-resolution image from the .ico file is used for that shortcut.

IOJ-1719380

The InstallShield prerequisites that install Microsoft SQL Server 2014 Express RTM (x86) and Microsoft SQL Server 2014 Express RTM (x86 & x64Wow) now download the correct files (SQLEXP_x86_ENU.exe and SQLEXP32_x86_ENU.exe, respectively). In addition, the conditions for these prerequisites have been revised so that the prerequisite files are launched on the appropriate platforms.

IOJ-1720047 (Advanced UI, Suite/Advanced UI)

If an Advanced UI or Suite/Advanced UI project includes an .msp patch that targets a large number of product codes, it is now possible to build a release in the project without encountering build error -7230 ("Unable to open MSP database *PathToMSP*").

IOJ-1720244 (Basic MSI)

The Setup.exe launcher no longer crashes in certain circumstances. Previously in some rare scenarios, when Setup.exe was performing cleanup tasks after, for example, a successful installation or an end user canceled, it crashed.

New Features in InstallShield 2014

Original Release Version (May 2014)

New InstallShield Prerequisites for the .NET Framework 4.5.1

InstallShield includes two new .NET Framework–related InstallShield prerequisites that you can add to Advanced UI, Basic MSI, InstallScript, InstallScript MSI, and Suite/Advanced UI projects:

- Microsoft .NET Framework 4.5.1 Full
- Microsoft .NET Framework 4.5.1 Web

These prerequisites install the .NET Framework 4.5.1 on supported target systems.

The Web prerequisite requires an Internet connection. This prerequisite downloads the required redistributable files if appropriate. The full prerequisite is a stand-alone installation that does not require an Internet connection.

Microsoft SQL Server 2014 Support

InstallShield now includes support for running SQL scripts on SQL Server 2014 database servers. In addition, InstallShield includes SQL Server 2014 in the predefined list of database servers that you can select when you are specifying in the SQL Scripts view the target database servers that your product supports.

If your installation targets SQL Server 2014, the SQLBrowse run-time dialog that is displayed when end users choose to browse for a database server can now list instances of SQL Server 2014, SQL Server 2014 Express, and SQL Server 2014 Express LocalDB. In addition, the SQLBrowse run-time dialog that is displayed when end users choose to browse for a database catalog can now list catalogs on the specified SQL Server 2014 database server.

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

Microsoft SQL Server 2014 Prerequisites

InstallShield includes several new SQL Server 2014–related InstallShield prerequisites that you can add to Advanced UI, Basic MSI, InstallScript, InstallScript MSI, and Suite/Advanced UI projects:

- Microsoft SQL Server 2014 Express RTM (x64)
- Microsoft SQL Server 2014 Express RTM (x86 & x64Wow)
- Microsoft SQL Server 2014 Express RTM (x86)

- Microsoft SQL Server 2014 Express RTM LocalDB (x64)
- Microsoft SQL Server 2014 Express RTM LocalDB (x86)

These InstallShield prerequisites install the technology on supported target systems.

New InstallShield Prerequisites for App-V 5.0 SP2

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 SP2 Desktop Client (x86)
- Microsoft App-V 5.0 SP2 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.

New InstallShield Prerequisites for Microsoft Visual C++ 2012 Update 4

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++ 2012 Update 4 Redistributable Package (x86)
- Microsoft Visual C++ 2012 Update 4 Redistributable Package (x64)

These prerequisites install the various technologies on supported target systems.

Support for Checking for Updates from Maintenance Mode of Advanced UI and Suite/Advanced UI Installations

Advanced UI and Suite/Advanced UI installations now have support for enabling end users to check for updates to your product by clicking a new Update button on the maintenance wizard page. If an update is available from your Web site and the end user chooses to obtain it, the installation downloads it, verifies its digital signature, and launches it.

To use this functionality, your Advanced UI or Suite/Advanced UI project must have support for automatic updates. At build time, InstallShield creates a metadata file called isupdate.xml file and places it in an UpdateMetadata subfolder in your project's build folder. This metadata file identifies the different versions of your Advanced UI or Suite/Advanced UI installation and their download locations. When you are ready to release the update for your product, you upload to your site this isupdate.xml file as well as your updated Advanced UI or Suite/Advanced UI installation.

At run time, if the end user is running maintenance mode, the Suite engine checks the metadata file for updates. If an update is available for download, the Suite engine sets the new `ISUpdateAvailable` property to 1 and displays a new `MaintenanceUpdateWelcome` wizard page, a maintenance wizard page that includes an Update button. In addition, the Suite engine sets the new `ISUpdateVersion` property equal to the version of the Suite/Advanced UI or Advanced UI installation that is available for download.

If an update is not available, the Suite engine displays the standard `MaintenanceWelcome` wizard page that does not have an Update button.

Ability to Include Support Files in Subfolders

InstallShield now lets you specify custom folder structures for support files. To add a subfolder under one of the language-specific nodes or to the language-independent node in the Support Files view, right-click the node and then click `New Folder`. InstallShield adds a subfolder and enables you to rename it as needed. You can also add a nested folder structure. To add files to a subfolder, select it and right-click in the right pane, and then click `Insert Files`. At run-time, the installation copies the support folders and files to `SUPPORTDIR`, a temporary directory on target systems, to make them available during your product's installation process. The support folders and files are deleted when the installation is complete.

The improved Support Files view is available in the following project types: Advanced UI, Basic MSI, Suite/Advanced UI, InstallScript, and InstallScript MSI.

The automation interface also includes subfolder functionality for support files. The `ISWiSetupFile` object includes a new read-write `Target` property that you can use to specify or obtain the path of the support file's target directory within `SUPPORTDIR`.

This feature resolves the following issues: IOA-000031237, IOC-000057723, IOC-000063377.

Support for Deploying Web Deploy Packages to IIS Web Servers and the Cloud

Suite/Advanced UI installations now have built-in support for deploying Web Deploy packages to IIS Web servers and the cloud. The Web Deploy packages can be created through IIS or a Web application development environment such as Visual Studio.

To add a Web Deploy package to your Suite/Advanced UI project, use the Packages view. When you add a package and select it in the Packages view, InstallShield displays package-related settings in the right pane. The Configuration area on the Common tab of the right pane includes several configuration-related settings for Web Deploy, such as the destination—which is often a remote server—and credentials. In addition, the Configuration area also includes settings for the parameters that are defined in the Web Deploy package's parameters XML file. By default, InstallShield uses Suite/Advanced UI properties for all of the configuration settings so that end users can set them at run time through the wizard interface or the command line.

When you add a Web Deploy package to your project, InstallShield adds a predefined Web Deploy wizard page to your project for that package. The Web Deploy wizard page enables the end user to specify whether the package will be deployed to a local IIS server, a remote server, or a cloud-based server. It also enables the end user to load the configuration settings from a publisher profile file (.publishsettings). If you want end users to be able to configure the parameters that are defined in your Web Deploy package's parameters XML file, you can add controls to the wizard interface as needed, and associate those controls with the properties that are specified for the corresponding parameter settings in the Packages view.

Note that Web Deploy packages are typically intended to behave similarly to first-time installations or to install over existing earlier or identical versions. As a side effect, Web Deploy packages do not handle uninstallation. Thus, Suite/Advanced UI installations do not offer maintenance for this type of package. It is recommended that you avoid creating a Suite/Advanced UI installation that combines one or more Web Deploy packages with one or more traditional packages (that do offer maintenance) as primary packages.

This feature is available in the Premier edition of InstallShield.

This feature resolves issue IOA-000070184.

Support for Including InstallShield Projects as Packages in Suite/Advanced UI Projects

InstallShield now lets you add InstallShield projects (.ism) as packages in Suite/Advanced UI projects. The InstallShield projects can be either Basic MSI projects or InstallScript projects; the projects must have been saved in the same version of InstallShield as the Suite/Advanced UI project.

When you start a build of a release for a Suite/Advanced UI project that includes one or more InstallShield project packages, InstallShield first builds the designated releases in the associated InstallShield projects, and includes them as packages in the Suite/Advanced UI installation that it generates. InstallShield adds the output of the Basic MSI and InstallScript projects to the Suite/Advanced UI installation as .msi and .hdr packages.

Use the Packages view to add an InstallShield project as a package in your Suite/Advanced UI project. This view is also where you identify which product configuration (for a Basic MSI project package) and which release (for both Basic MSI project packages and InstallScript project packages) that you want to include as a package in the Suite/Advanced UI project.

This feature is available in the Premier edition of InstallShield.

Support for InstallScript Actions in Suite/Advanced UI Installations

Suite/Advanced UI installations now have built-in support for launching InstallScript actions to perform various run-time tasks that are outside the scope of the packages that you are including in the installation. These Suite/Advanced UI-based InstallScript actions function much like InstallScript custom actions in Basic MSI projects.

You can use the InstallScript view that is available now in Suite/Advanced UI projects to add InstallScript files (.rul) to your project and write functions using the InstallScript language. To have your Suite/Advanced UI action call an InstallScript function, you must prototype the function with the export keyword, just as you would for InstallScript custom actions in Basic MSI projects.

Once you have added InstallScript code to your project, use the Events view to add an InstallScript action that calls your code. Next, you can schedule when at run time you want the action to be launched. For example, you can use the Events view to schedule an InstallScript action to run during one or more of the built-in events that the Suite/Advanced UI engine manages.

Note that there are some limitations to the use of InstallScript in Suite/Advanced UI projects. For example, InstallScript UI-related functions and InstallScript run-time path variables (such as PROGRAMFILES and WINDIR) are not available for use in a Suite/Advanced UI project.

This feature is available in the Premier edition of InstallShield.

Support for Managed-Code Actions in Suite/Advanced UI Installations

Suite/Advanced UI installations now have built-in support for launching managed-code actions to perform various run-time tasks that are outside the scope of the packages that you are including in the installation. The new action type calls a public method in a .NET assembly that is written in managed code such as Visual Basic .NET or C#. These Suite/Advanced UI-based InstallScript actions are functionally equivalent to DLL actions in Suite/Advanced UI installations. That is, Suite engine looks for one parameter on the managed function that it calls. This parameter is an ISuiteExtension interface passed as a System.Object. The return type should be of some 32-bit integer type.

To add a managed-code action to a Suite/Advanced UI project, use the Events view.

This feature is available in the Premier edition of InstallShield.

Ability to Launch Suite/Advanced UI Event Actions that End Users Trigger When Using Wizard Interface Controls

Suite/Advanced UI installations include support for run-time actions that are defined in the Events view and that are launched when an end user clicks or performs some other action on a wizard interface control. For example, this capability enables your installation to run executable files, call DLL

functions, run InstallScript code, call managed methods in a managed assembly, run PowerShell scripts, or set a Suite/Advanced UI property when an end user clicks a button in the wizard interface.

To add an action to your project, use the Events view.

To schedule an action to run when a wizard interface control is clicked or used, select the control that you want to configure in the Wizard Interface view, and then use the Click setting or an action-related setting under the Events setting to select the action that you want to launch.

This feature is available in the Premier edition of InstallShield.

This feature resolves issue IOA-000085159.

Ability to Launch Suite/Advanced UI Actions When Wizard Pages and Windows Are Shown or Hidden

Suite/Advanced UI installations include support for run-time actions that are launched when a wizard page or secondary window is opened or closed. This capability enables your installation to perform any initialization such as dynamically retrieving values for a combo box. It also lets you schedule actions to occur before the wizard interface is displayed at run time.

To add an action to your project, use the Events view.

To schedule an action to run when a wizard page opens or closes, select the wizard page that you want to configure in the Wizard Interface view, and then select the appropriate action in the Page Entered setting or the Page Exited setting.

To schedule an action to run when a secondary window opens or closes, select the secondary window that you want to configure in the Wizard Interface view, and then select the appropriate action in the Window Entered setting or the Window Exited setting.

This feature is available in the Premier edition of InstallShield.

New Property for Preventing Passwords from Being Written in Advanced UI and Suite/Advanced UI Installation Log Files

If you launch an Advanced UI or Suite/Advanced UI Setup.exe file with the /debuglog parameter, the Suite engine generates a debuglog file. By default, the debug log file includes the values of changed Advanced UI or Suite/Advanced UI properties.

In some cases, you may want to prevent the Suite engine from writing the values of specific properties to the debug log file. For example, you may want to prevent properties that contain passwords and other sensitive information from being logged. To enable this scenario, Advanced UI and Suite/Advanced UI installations support a new property called ISHiddenProperties. You can set this property to a semicolon-delimited list of case-sensitive property names whose values you do not want to be written to debug log files.

Note that ISHiddenProperties is useful for prevention of logging only for values that would be logged by changing the property value. If the property is logged any other way (such as through ISuiteExtension::LogInfo), the Suite engine cannot prevent the logging. Therefore, any code that you create to write a property value to the log file should read ISHiddenProperties to see whether the property value should be logged.

This feature resolves issue IOA-000064725.

Enhanced Substitution Syntax for Advanced UI and Suite/Advanced UI Projects

Various areas of Advanced UI and Suite/Advanced UI projects include support for enclosing property names in square brackets to indicate that the properties should be replaced with appropriate values at run time. The run-time syntax has been enhanced to support several additional types of substitution:

- **Special characters**—Preceding a special character with a backslash and enclosing the resulting string in square brackets resolves the special character at run time. For example, an entry such as `[\\Text\\]` resolves as `[Text]`.
- **Null character**—A tilde that is enclosed in square brackets—that is, `[~]`—resolves as a null character embedded in the string.
- **Environment variables**—Preceding the name of an environment variable with a percent sign and enclosing the resulting string in square brackets resolves the environment variable at run time. For example, `[%PATH]` resolves as the value of the PATH environment variable.
- **Recursive property resolution**—Surrounding a property resolution with additional square brackets resolves the resolved value of a property. For example `[[PROPERTY1]]` resolves to the value of the property whose name is stored in **PROPERTY1**; if **PROPERTY1** holds **PROPERTY2**, the resolved value is the value stored in **PROPERTY2**.

Support for Removing Files and Folders

InstallShield now has built-in support that makes it easy to specify files and folders that you want to be removed from target systems at run time. This file and folder removal capability is useful for scenarios such as removing application-created files that your installation does not otherwise track.

You can schedule the removal of a file or folder for one of the following events:

- When the file or folder's component is being installed
- When the file or folder's component is being uninstalled
- When the file or folder's component is being installed or uninstalled

Note that if the item to be removed is a folder, that folder is removed only if it is empty.

InstallShield offers different methods for configuring a file or folder removal in a project:

- In the Files and Folders view, select the destination folder that contains the file or folder that you want to be removed. Then right-click in the **Destination computer's files** pane and click Add File Removal.
- In the Setup Design view (in installation projects) or the Components view, expand the node of the component that contains the file or folder that you want to be removed, and then click the Files subnode. Next, right-click in the Files pane and click Add File Removal.

With both methods, InstallShield displays a Properties dialog box that lets you configure the available removal settings.

Previously, the only way to indicate which files and folders should be removed was to use the Direct Editor view to manually author entries in the RemoveFile table.

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

The automation interface includes a new ISWiRemoveFile object and a new ISWiRemoveFiles collection for removing files and folders. In addition, three new member methods are applicable to the ISWiComponent object: AddRemoveFile, which enables you to add a file removal entry to the current component; RemoveRemoveFile, which enables you to remove a file removal entry from the current component; and ISWiRemoveFiles, which returns an ISWiRemoveFiles collection that contains all of the file removal entries that are associated with the current component.

These automation interface improvements are available in the following project types: Basic MSI, InstallScript MSI, and Merge Module.

This feature resolves the following issues: IOA-000055322 and IOC-000047734.

Support for DPI-Aware Installations

InstallShield now lets you create DPI-aware installations that can scale run-time user interface elements properly across modern target systems that have various high-DPI display settings, making your installation's user interface more consistent and visually appealing. The support that is available and the improvements that have been made vary, depending on the project type that you are using.

Advanced UI and Suite/Advanced UI Projects

DPI-awareness in Advanced UI and Suite/Advanced UI projects has two parts:

- Items that are drawn by the engine are scaled according to the target system's DPI settings. Thus, if a target system has a DPI of 200%, the check box control is scaled accordingly.
- The engine considers the scale factor and language when displaying images and other resources that you are including in the wizard interface.

For example, when the engine determines that the scale factor should be 150 and that the appropriate language is English, it looks for resources that have the string *scale-150* in their path or file name. It searches the following paths in the order listed when searching for *image.png* on the target system:

1. [SUPPORTDIR]0409\scale-150\image.png
2. [SUPPORTDIR]0409\image.scale-150.png
3. [SUPPORTDIR]0409\image.png
4. [SUPPORTDIR]scale-150\image.png
5. [SUPPORTDIR]image.scale-150.png
6. [SUPPORTDIR]\image.png

Note that matching the language takes precedence over matching the correct DPI.

InstallShield includes *scale-150* and *scale-200* images for the built-in default images that are included in Advanced UI and Suite/Advanced UI projects. If you want to include custom resources in your project, use the Support Files view to add appropriately scaled images.

InstallScript and InstallScript MSI Projects

DPI-awareness in InstallScript and InstallScript MSI projects has the following parts:

- All of the built-in default dialog images have been replaced with images that scale well to 200%.
- The static dialog controls that have a control ID of 1200 now include support for .png images, which allow for transparency. This applies to the computer image that is displayed by default on the left side of the exterior skinned dialogs such as the Welcome dialog. It also applies to the banner images on interior dialogs.

In addition, these same controls support a new format for identifying each image and its associated scale factor:

`@@ResourceID;ScaleFactor`

`ResourceID` indicates the resource identifier number that should be used to look up either a .png image (stored as resource type PNG) or a bitmap image (stored as resource type RT_BITMAP). `ScaleFactor` indicates the DPI scale percentage for which the image is intended.

For example, this can be 100% (96 DPI), 125% (120 DPI), 150% (144 DPI), or 200% (192 DPI). If the scale factor that is specified for an image is 200, the image will be shrunk down for display on target systems that are running less than 200% DPI scaling. On a 200% target system, it will be displayed at 1:1. If the scale factor that is specified for the image is 100, the image will be scaled up for display on target systems that are running 200% DPI scaling.

The previous format for identifying bitmap images (.bmp) still works, but it does not have

support for scaling, or for .png images:

```
@BitmapResourceID;TransparentFlag;3-DFlag;<unused>;TransparentColorKey
```

- The new image string format is now used by default on all ID 1200 static controls for the dialog header banner and for exterior panel icon images. If you upgrade an InstallScript or InstallScript MSI project to InstallShield 2014 and you edited the dialogs in the earlier version of InstallShield, you will need to edit the image string to use the new format for all of the languages that your project supports.
- The InstallScript function SdOptionsButtons has been updated to include support for the new string format for identifying images:

```
@@ResourceID;ScaleFactor
```

Previously, the following format was used:

```
@BitmapResourceID;BitmapIconFlag;TransparentColor
```

- The nItem parameter for the InstallScript function GetSystemInfo includes support for two new constants:
 - SYSTEM_DPI—This returns the system DPI value in the nvResult parameter.
 - SYSTEM_DPI_SCALING—This returns the system DPI scaling value in the nvResult parameter.

Basic MSI Projects

The built-in images that are displayed on dialogs that the Setup.exe launcher displays at run time have been replaced with images that scale well to larger areas. This includes images that are displayed on the initialization dialog.

This feature resolves the following issues: 1-1770V5, IOA-000070081, IOA-000078637.

Support for Distributing Your Installation to Virtual Machines that InstallShield Provisions at Build Time or on Demand

You can configure your projects so that after each successful build of your installation, InstallShield automatically reverts a virtual machine (VM) to a designated snapshot, powers on the VM, and copies your installation to the VM to make it available for testing. You can also alternately perform these testing preparation steps on demand at any time. The testing preparation capability makes it possible to reduce testing time and eliminate manual steps.

The VM can be on a Microsoft Hyper-V Server, a VMware ESX or ESXi Server, or a VMware Workstation.

To specify VM information, use the Virtual Machine area on the Events tab for a release in the Releases view. One of the settings lets you specify whether you want the release to be distributed to the specified VM each time that it is successfully built.

To distribute the release to the VM at any time, right-click the release in the Releases view and then click Distribute to VM.

Note that when you use the Configuration setting on the Events tab to configure VM details, InstallShield writes the data that you specify to a file called VMConfigurations.xml. This VMConfigurations.xml file is a machine-wide file that you can configure once and then use it in other InstallShield projects, as well as share it with other team members. You can also use this file with the Standalone Build.

In order to deploy to one of the supported VMs, InstallShield needs to communicate with the virtualization technology that you are using. If you are using VMware virtualization technology (VMware ESX or ESXi Server or VMware Workstation), the VMware VIX API needs to be installed on the same machine as InstallShield. You can do this by either installing VMware Workstation on that machine or by downloading and installing the VMware VIX API from <http://www.vmware.com/support/developer/vix-api>.

If you are using VMware Workstation, it is recommended that you install VMware Workstation on the same machine as InstallShield so that InstallShield uses the version of the VIX API that was designed for that specific version of VMware Workstation. Although it is likely that newer versions of the VIX API will also work, it seems that the best approach is for InstallShield to use the version of the VIX API that was bundled with your version of VMware Workstation.

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

This feature is available in the Premier edition of InstallShield.

Enhancements in InstallShield 2014

Original Release Version (May 2014)

Usability Improvements for Populating Combo Box and List Box Controls in Advanced UI and Suite/Advanced UI Projects

InstallShield lets you add combo box controls and list box controls to wizard pages and secondary windows of Advanced UI and Suite/Advanced UI projects. The process of creating either one of these types of controls involves the use of two properties: one that defines the list of options that you want to display in the control, and one that the installation uses to store the option that the end user selects at run time. Now you can configure both of those properties entirely from within the Wizard Interface view in InstallShield, instead of having to use the Wizard Interface view and the Property Manager view.

In the Wizard Interface view, the Content Property setting of the combo box or list box control now contains an ellipsis button (...) that you can click when you want to define the list of options that you want to display in the control. When you click this button, a simple Edit List Options dialog box opens, enabling you to specify the list of options that you want to display in the selected control, as well as an associated value.

Previously, it was necessary to use the Property Manager view to define the property that contained the list of options, as well as their corresponding property values. The format that was required for defining the property (for example, ID_Option1\rValue1\nID_Option2\rValue2\nID_Option3\rValue3) was difficult to remember and error-prone.

Run-Time Improvements for Browsing to a File or Folder in Advanced UI and Suite/Advanced UI Installations

Several improvements for enabling end users to browse to a file or folder are now available in Advanced UI and Suite/Advanced UI installations:

- A new Browse for New File action is available for controls on wizard pages and secondary windows. If you add this action to the Click event of a control such as a button and an end user clicks the button, a browse dialog box opens, enabling the end user to specify a new file and its location.

When you add this type of action to a control in your project, you can specify the text for the title bar of the dialog box. You can also specify the file extension filter options that you want to be available for selection, as well as the default file extension that should be used when creating the file.

- The existing Browse for Folder action has been enhanced. The Browse For Folder dialog box that this action launches accepts only locations that have file system equivalents. That is, the OK

button on this dialog box remains disabled if the end user selects a location such as This PC or My Computer. Previously, end users were able to select such locations, and doing so cleared the value of the associated property.

When you add this type of action to a control in your project, you can specify the instructions that you want to be displayed on this Browse For Folder dialog box.

- The existing Browse for File action has been enhanced to enable you to specify the file extension filter options that you want to be available for selection on the file browse dialog box.

Support for Offering Printer Selection for the LicenseAgreement Dialog at Run Time

The behavior of the Print button on LicenseAgreement dialogs has been enhanced. Instead of printing directly to the default printer when an end user clicks the Print button, the printer selection dialog box now opens.

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

Advanced UI and Suite/Advanced UI projects already have this capability.

This enhancement resolves issue IOC-000047168.

Ability to Insert Text in Text Files

The Text File Changes view enables you to configure changes that you want to be made to text files at run time on target systems. The text files can be part of your installation, or they can be files that are already present on target systems. The Text File Changes view has been improved to enable you to insert text at the beginning or end of one or more files. It also now lets you insert text before or after specified text. When you are configuring text file changes in this view, use the new Action setting to specify whether you want to replace existing text, insert text before or after existing text, or insert text at the beginning or end of files.

Previously, the Text File Changes view had support for replacing existing text in text files at run time.

This enhancement is available in the following project types: Basic MSI, DIM, InstallScript MSI, Merge Module, MSI Database, Transform.

Support for Enabling End Users to Choose to Which Installed Instances of an Earlier Version of a Product to Apply a Major Upgrade

The multi-instance support in Basic MSI projects now includes support for major upgrades. At run time, the instance selection dialog that the Setup.exe setup launcher displays lets end users view a list of all of the currently installed instances of the product, including different major versions of the

product that are installed. End users can now use this improved instance dialog to install a new instance, maintain an existing instance, or apply a major upgrade for an instance.

Previously, the only instances that the instance selection dialog listed were ones that shared the same product code as the running multi-instance installation.

This enhancement resolves the following issues: IOC-000076675 and IOC-000077274.

Ability to Create Microsoft App-V 5.x Packages on Additional Platforms

The Microsoft App-V Assistant in InstallShield includes support for creating virtual applications in the Microsoft App-V 5.x format. The Package Information page in this assistant lets you specify which version of App-V—5.x or 4.x—you want to target. Now you can build an App-V 5.x package in InstallShield on any version of Windows that InstallShield supports. Previously, Windows 8 or Windows Server 2012 was required.

Enhanced File Open Run-Time Dialog for Basic MSI Installations

InstallShield includes support for launching the Open dialog from one of the dialogs in your Basic MSI installation. End users click a browse button in one of your dialogs, and this launches the Open dialog. The Open dialog lets the end user browse for a file. You can now optionally use the new property IS_BROWSE_INITIALDIR to specify the default path that is displayed in this dialog at run time. Previously, the dialog displayed the path to which the end user last browsed.

This enhancement resolves issue IOA-000083510.

New FlexNet Connect 13.06 Redistributables Available

InstallShield includes support for FlexNet Connect 13.06 in Basic MSI and InstallScript MSI projects. Use the Update Notifications view in InstallShield to include one of the two FlexNet Connect 13.06 merge modules—one has the Common Software Manager, and the other does not.

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.

Concurrent Licensing Changes

If you purchase concurrent licenses of InstallShield, InstallShield Collaboration—also known as InstallShield Developer Installation Manifest (DIM) Editor—or the Standalone Build, a new version of the FlexNet Licensing Server software is required for the licensing server that you set up to manage your organization's licenses. This software includes improvements that are not available in earlier versions of the software.

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

You can obtain the installations of InstallShield and the Standalone Build 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 and the Standalone Build.

Installing More than One Edition of InstallShield

Only one edition of InstallShield 2014—Premier, Professional, or Express—can be installed on a system at a time. In addition, the InstallShield 2014 DIM Editor cannot be installed on the same machine with any edition of InstallShield 2014.

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 2014 can coexist on the same machine with other versions of InstallShield.

The InstallShield 2014 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 2013 and earlier to InstallShield 2014. It also alerts you to possible changes in behavior that you may notice between new InstallShield 2014 projects and projects that are upgraded from InstallShield 2013 or earlier to InstallShield 2014.

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

If you use InstallShield 2014 to open a project that was created with an earlier version, InstallShield 2014 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 .774 (for an .ism project) or .2013 (for an .issuite project) before converting it. Delete the .774 or .2013 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 2014 projects in earlier versions of InstallShield.

You can upgrade projects that were created with the following versions of InstallShield to InstallShield 2014: InstallShield 2013 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 2014.

Deprecation of One-Click Install Installations

One-Click Install installations have been deprecated in favor of Suite/Advanced UI installations. In a future release, the ability to create and build One-Click Install installations will be removed from Basic MSI, InstallScript, and InstallScript MSI projects. That is, the Generate One-Click Install settings for a release in the Releases view will no longer be available, and InstallShield will no longer generate an installation whose initial user interface is an HTML page.

Various alternatives for One-Click Install installations are available. For example, you can create your own Web page with a link to your installation, and host the page and your installation on your organization's Web site. You can use a solution such as FlexNet Operations, available from Flexera Software, to manage electronic delivery and optionally licensing of your product.

You can also consider using Suite/Advanced UI installations, which have support for packaging together installations as a single installation while providing a unified, fully customizable user interface. In addition, you can configure Suite/Advanced UI installations to download only the packages that are required to each specific target system at run time.

Automation Interface Changes

If you use the automation interface with InstallShield or the Standalone Build, update your existing code to reflect the new ProgID: IswiAuto21.ISWiProject. The Standalone Automation Interface uses the same ISWiAutomation21.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 ISWiAutomation21.dll file that is registered is the one that is used.

Resolved Issues in InstallShield 2014 Original Release Version (May 2014)

IOA-000058026 (Basic MSI, DIM, Merge Module)

The Dialog Editor no longer discards conditions that are entered for multiple actions in the Behavior area of a dialog in the Dialogs view. Previously under some scenarios, the conditions that were entered were not saved.

IOA-000058903 (Basic MSI, DIM, InstallScript MSI, Merge Module)

If you include adodb.dll as a reference in a Visual Studio solution that includes an InstallShield project, adodb.dll is now included in the installation that InstallShield builds. To resolve this issue, adodb.dll was removed from the Filters.xml file that is installed with InstallShield.

IOA-000059752 (Basic MSI, DIM, InstallScript, InstallScript MSI, Merge Module)

If you use the Custom Errors setting for a Web site in the Internet Information Services view to configure one or more custom HTTP errors, InstallShield no longer adds a <clear /> element to the web.config file on target systems.

IOA-000061496 (Basic MSI, DIM, InstallScript, InstallScript MSI, Merge Module)

By default, the value of the string identifier IDS_ERROR_130 no longer references a specific minimum required version of IIS. Previously, the default value of the string referenced IIS 4.0.

Note that this string change occurs for new projects that are created in InstallShield 2014 or later. If you upgrade an InstallShield 2013 or earlier project that contains IIS support to InstallShield 2014, the value of this string is not updated automatically. You can modify the value of the string in the String Editor view.

IOA-000062113 (Basic MSI, DIM, InstallScript MSI, Merge Module)

The built-in COM+ support has been modified to use the ISComponentServiceUninstall custom action, a deferred execution in system context action, to uninstall COM+ applications. This change makes it acceptable to select Invoker for the Required Execution Level setting in the Releases view for a release that includes a COM+ application. Note if the uninstallation fails, the COM+ changes are no longer rolled back during uninstallation.

Previously, if the Required Execution Level setting in the Releases view was set to Invoker, run-time error 27509 occurred when the ISComponentServiceFinalize custom action, which was previously used to uninstall COM+ applications, tried to uninstall the COM+ application.

IOA-000063553 (MSI Database)

When a merge module that includes a COM+ application is added to a Basic MSI project and the **Install after InstallFinalize action** check box is selected for the COM+ application, the **Install after InstallFinalize action** check box is now selected in the Component Services view in the following scenario: the MSI database that is built from the project is opened in InstallShield in Direct Edit mode.

IOA-000068405, IOA-000077563 (Basic MSI, InstallScript MSI)

If you specify a digital signature password for a patch configuration in the Patch Design view, and if the file format that you are using for your project is XML, InstallShield now encrypts the password in the XML file. Therefore, if the XML project file is opened in a text editor, the password is not displayed as plain text.

IOA-000073809 (Advanced UI, Basic MSI, InstallScript, InstallScript MSI, Suite/Advanced UI)

If you want to use a custom icon for your Setup.exe file, you can now specify in the Setup.exe Icon File setting on the Setup.exe tab in the Releases view an icon file that uses a non-numeric resource ID. Previously, if you specified a non-numeric resource ID, build warning -7212 occurred at build time, and InstallShield used the default icon instead of your custom icon.

IOA-000076595 (Advanced UI, Suite/Advanced UI)

Suite/Advanced UI and Advanced UI installations have two new feature settings:

- **Allow UI Selection Change**—Use this setting to indicate whether end users should be allowed to select or deselect the feature on the InstallationFeatures wizard page. The default value is Yes, indicating that end users are allowed to deselect the feature if they do not want to install it.
- **Follow Parent State**—Use this setting to indicate whether you want the selected subfeature's state to be determined by its parent feature. The default value is No, indicating that the subfeature may remain uninstalled when its parent is installed.

IOA-000078358 (Suite/Advanced UI)

Build error -7242 no longer occurs if you are building a Suite/Advanced UI installation that has an .exe package, and if target file is selected in at least one of the Target subsettings in the Operations area for that .exe package in the Packages view. Previously, whenever the Target subsetting under the Install operation was null for an .exe package, build error -7242 occurred.

IOA-000078470 (Basic MSI)

The Setup.exe launcher now correctly resolves the Windows Installer properties SETUPEXEDIR and SETUPEXENAME for feature prerequisites in silent installations. Previously, the SETUPEXEDIR property was set to an empty string at run time, and attempts to use this property in a feature prerequisite's command-line parameters failed.

IOA-000078765 (MSP, Transform)

InstallShield now successfully opens patches (.msp files) and transforms (.mst files). Previously in some scenarios, InstallShield displayed an error when attempting to open certain .msp and .mst files.

IOA-000079975

The example in the "SELECTED_LANGUAGE" help topic in the InstallShield Help Library is now correct. It now lists an ID of ISLANG_ENGLISH_UNITEDSTATES for the text substitution value of 0x0409. Previously, this help topic listed an ID of ISLANG_ENGLISH.

IOA-000080617, IOA-000081810

If you select a grid item in an area such as the Property Manager view or the Direct Editor and then change the focus, the selected item now uses a darker color that makes it easier to identify it as the selected item.

IOA-000081151

If you select DVD-5 for the Media Format setting of a release on an English system and then build the installation on a machine that has a different system locale, a "type mismatch" error no longer occurs, and build error -5036 no longer occurs.

IOA-000081185 (Advanced UI, Basic MSI, InstallScript, InstallScript MSI, Suite/Advanced UI)

The existing InstallShield prerequisite for Microsoft SQL CE 3.5 SP2 was revised to install SQL CE 3.5 SP2 on 64-bit target systems; previously, the prerequisite's conditions were configured to install SQL CE 3.5 SP2 only on 32-bit systems. In addition, a new InstallShield prerequisite for Microsoft SQL CE 3.5 SP2 (x64) is available; this prerequisite installs SQL CE 3.5 SP2 on 64-bit target systems.

IOA-000081812 (Advanced UI, Suite/Advanced UI)

If you configure an InstallShield prerequisite to have a condition of "A file with a certain version exists" and you select the option "If the file's version is LESS THAN the version specified above (or the file IS NOT FOUND on the target system)," and then you add this prerequisite to an Advanced UI or Suite/Advanced UI project, the conditions for that prerequisite package are now configured correctly.

IOA-000082248 (Basic MSI, DIM, InstallScript MSI, Merge Module)

If you use the Reg-Free COM Wizard to configure REG-free COM manifest files for your project, InstallShield creates the proper manifest file and adds the manifest component to your project without displaying an error. Previously in some cases, InstallShield displayed an error and created an empty manifest file.

IOA-000082406

InstallShield no longer crashes when certain files such as msvcrt.dll are added to a project.

IOA-000082730 (InstallScript)

It is now possible to run a compressed InstallScript installation that has a media format of network image and that is 2 GB or larger. Previously, the installation displayed an error stating that the setup launcher stopped working, and the installation failed.

Note that Windows-based systems have a size limit of 4 GB for loading files, so installation files should not exceed this limit.

IOA-000082864 (Basic MSI)

If you add a DIM reference to a Basic MSI project, and the DIM has one or more custom actions, it is now possible to edit the DIM's sequence settings in the Custom Actions and Sequences view.

IOA-000083008 (Advanced UI, Suite/Advanced UI)

If the wizard interface includes a billboard control that includes a mode condition for its Visible setting, the billboard control is now hidden when expected at run time. Previously, the billboard control was visible in all cases.

IOA-000083044 (Advanced UI, Suite/Advanced UI)

If you specify a maximum service pack number (such as -2, or another value that starts with a dash) in a platform condition in your project, the installation now evaluates the condition properly at run time. Previously, the installation failed with error 0x80070057 (An invalid condition was encountered).

IOA-000083382 (Basic MSI, InstallScript MSI)

If you include special characters such as a period in the name of a component, you select the Install from the Web option for your release, and you select the one .cab file per component option, the installation can now successfully download and install the files of that component. Previously, the installation failed with a 1606 error because the file name of the .cab file was different than the name of the file that the installation attempted to download.

IOA-000083658 (Basic MSI, DIM, InstallScript MSI, Merge Module, Transform)

If you have a service in the Services view that includes escaped square brackets in its name, InstallShield now enables you to rename the service to add characters before or after the escaped square brackets. Previously in this scenario, InstallShield displayed an error stating that the service name contained illegal characters.

IOA-000084390 (Advanced UI, Suite Advanced UI)

If you have an InstallShield prerequisite such as the Microsoft Visual C++ 2005 SP1 Redistributable Package (x86) prerequisite, which has a registry condition that checks the 64-bit area of the registry, and you add this prerequisite to an Advanced UI or Suite/Advanced UI project, the registry condition for that prerequisite package is now configured correctly.

IOA-000084391

The conditions for the InstallShield prerequisites for the Microsoft Visual C++ 2012 Update 1 Redistributable Packages have been updated to include support for Windows XP. Previously, these prerequisites were not installed on Windows XP-based systems.

IOA-000084479 (Advanced UI, InstallScript, Suite/Advanced UI)

When Advanced UI and Suite/Advanced UI installations evaluate platform conditions on Windows 8.1 and Windows Server 2012 R2 target systems, the conditions are now evaluated correctly. Previously, Windows 8.1 systems were erroneously considered to be Windows 8 systems, and Windows Server 2012 R2 systems were erroneously considered to be Windows Server 2012 systems.

When the SYSINFO member nISOSL is used in InstallScript installations, it is now set to ISOSL_WIN81 on Windows 8.1 and Windows Server 2012 R2 target systems; previously, it was set to ISOSL_WIN8, which is the value that should apply only to Windows 8 and Windows Server 2012 systems.

IOA-000084687 (Basic MSI, InstallScript MSI)

If your installation includes a chained .msi package that returns ERROR_SUCCESS_REBOOT_REQUIRED (3010), the main installation no longer treats the 3010 return code as an error that leads to a rollback.

IOA-000084831

The documentation that describes the Template Summary property has been corrected. The documentation now indicates that if the platform portion of this property is left empty (for example, the Template Summary property is set to ;1036), the target processor is Intel. Previously, the documentation erroneously stated that the package is processor independent in that scenario.

IOA-000084832 (Advanced UI, Suite/Advanced UI)

When an Advanced UI or Suite/Advanced UI installation is installing a package, the name of the package is now included in the status message (for example, "Installing Package MyPackage"). Furthermore, at the appropriate place in the debug log file, the ISInstallStatus property is set to a

value that includes the name of the package (for example, "Installing Package MyPackage"). Previously, the package name was not included in these strings.

IOA-000084869 (Basic MSI, InstallScript MSI, Merge Module, Transform)

If you leave the platform portion of the Template Summary property empty (for example, the Template Summary property is set to ;1033), InstallShield now sets the Template Summary property in the .msi package that it creates at build time correctly (for example, ;1033), and end users can launch the installation without error on Intel platforms. Previously, if you left the platform portion of this property empty, InstallShield repeated the language identifier in the value of the Template Summary property in the .msi package that it created at build time (for example, 1033;1033). As a result, when end users launched the installation on Intel platforms, the error message "This installation package is not supported by this processor type" was displayed, and the installation failed.

IOA-000085115 (Advanced UI, Suite/Advanced UI)

If you specify a custom executable file name for your setup launcher and you specify digital signature information for your release, the custom name that you specified is now used in the UAC prompt and in the properties of your setup launcher file. Previously in this scenario, the default value of **setup** was used in the UAC prompt and in the properties of your setup launcher file.

IOA-000085184 (Suite/Advanced UI)

If you include a call to the InstallScript function LaunchApplication (or one of its earlier variants) in the InstallScript view of your Suite/Advanced UI project, InstallShield no longer encounters compile and link errors for OnLaunchAppAndWaitCallback and SdShowMsg at build time.

Note that as part of the fix for this issue, InstallScript actions in Suite/Advanced UI projects no longer support the structure member LAAW_PARAMETERS.szStatusText.

IOA-000123970 (InstallScript, InstallScript MSI)

If an InstallScript dialog in your project includes an HTML control that has a hyperlink with target="_blank" code, clicking the control at run time now opens the appropriate web page on systems that have Internet Explorer 10 or later. Previously, nothing happened when end users clicked this type of control.

IOA-000124405 (Advanced UI, Suite/Advanced UI)

If you use release flags to exclude a package or subfeature from your build, InstallShield now fully excludes the package, as well as packages that belong to subfeatures that this excludes. Previously,

the packages or the references to the packages could remain behind, resulting in build-time errors or run-time messages in the log.

IOA-000124532

If you specify .spc and .pvk files to digitally sign your release at build time, and a different application gains focus at build time, the password is passed correctly to Signcode.exe (the tool that digitally signs files as needed at build time). Previously, the password was written to the other application, and the Signcode.exe prompt for the certificate password remained open.

IOA-000124575 (Basic MSI, InstallScript, InstallScript MSI)

If you set the Splash property of the ISWiSetupFile object of the automation interface to true, the support file is now configured to be a splash screen. Previously, the support file was not configured correctly, and the support file was not used as a splash screen at run time in the resulting build.

IOA-000124576 (InstallScript, InstallScript MSI)

If the destination of one or more components in a project is set to WINSYSDIR, the mouse pointer no longer flickers when the end user selects and clears feature check boxes on the SdFeatureTree dialog at run time. Previously, this behavior occurred on some 64-bit target systems.

IOA-000124867 (Advanced UI, Suite/Advanced UI)

Displaying a wizard page no longer overrides a radio button selection when the first tab stop on the page is not the selected radio button. Previously, when a radio button was the first tab stop, displaying the wizard page would select that radio button.

IOB-000063651 (Basic MSI, DIM, InstallScript, InstallScript MSI, Merge Module, Transform, QuickPatch)

Exporting a registry entry that has an expandable string value to a .reg file and then importing it back into your project no longer results in unexpected registry entries.

IOC-000056252 (InstallScript, InstallScript MSI)

The built-in Japanese run-time dialogs no longer have overlapping Back and Next buttons; the Back buttons have been moved one dialog unit to the left. Previously, some of the Japanese dialogs, such as the Finish dialog, had buttons that overlapped slightly.

Note that this change does not apply to Japanese dialogs that have already been manually edited in InstallShield. To resolve the issue for those dialogs, it is necessary to manually edit the dialogs and move the buttons as needed.

IOC-000089095

InstallShield no longer erroneously reports build error -1007 ("Cannot copy source 'NameOfFile' to target 'FileLocation'") for an InstallShield prerequisite whose source is in a read-only location. Previously, this build error occurred in certain cases; for example, when Compressed was selected for the Compression setting in the Releases view and the target location was not Extract From Setup.exe.

IOC-000089170 (Basic MSI, InstallScript MSI)

If an InstallShield project is part of a Visual Studio solution that is configured in Team Foundation Server to use a drop folder (in which the build output is copied to a network folder), and if Yes is selected for the Shallow Folder Structure setting in the Releases view of the InstallShield project, the InstallShield build output is now copied to the drop folder as expected.

IOC-000090841 (Advanced UI, Suite/Advanced UI)

The ISReleaseFlags property is now set at build time to a list of release flags that are included in the release. You can use this property in conditions as needed.

IOC-000090875 (Basic MSI, InstallScript MSI)

If error 1152 occurs at run time when Setup.exe encounters an error when extracting a file to a temporary location (for example, if the target system does not have enough space), the error message now lists the name of the file that could not be extracted.

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

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 2008

Visual Studio 2010

Visual Studio 2012

Visual Studio 2013

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 (Windows XP SP3 for Advanced UI and Suite/Advanced UI installations)

Windows Server 2003 (Windows Server 2003 SP2 for Advanced UI and Suite/Advanced UI installations)

Windows Vista

Windows Server 2008

Windows 7

Windows Server 2008 R2

Windows 8

Windows Server 2012

Windows 8.1

Windows Server 2012 R2

Target systems must also support the SSE2 instruction set.

Known Issues

For a list of known issues, see Knowledge Base article [000017081](#).

Legal Information

Copyright Notice

Copyright © 2014 Flexera Software LLC. All Rights Reserved.

This publication contains proprietary and confidential information and creative works owned by Flexera Software LLC and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such publication in whole or in part in any form or by any means without the prior express written permission of Flexera Software LLC is strictly prohibited. Except where expressly provided by Flexera Software LLC in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera Software LLC intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera Software LLC, must display this notice of copyright and ownership in full.

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.