InstallShield®

WHITE PAPER

Validating MSI Updates and Patches

by Robert Dickau
Principal Technical Training Writer, Acresso Software

Acresso™ SOFTWARE
Powering the business of software

Validating MSI Updates and Patches

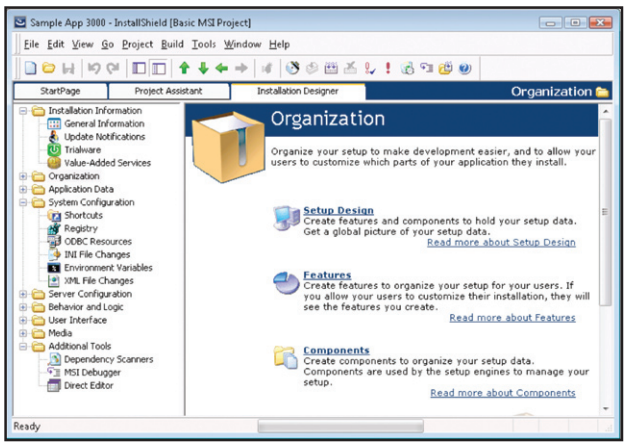
Introduction

The success of a Windows Installer–based update package involves a long list of considerations when the update is being performed. These considerations involve file versioning, key file information, product and component GUIDs, and more, and differ according to the update type (small update, minor upgrade, or major upgrade) and packaging (full installation or patch package).

InstallShield supports different validation rules for upgrades and patches, which can assist you in identifying and avoiding potential problems. This white paper describes these built-in validation rules, and how to address any validation warnings or errors that occur.

Using the InstallShield Environment

This white paper frequently refers to the InstallShield development environment. It is assumed you are familiar with the general layout of the InstallShield interface, which contains a list of views with which you can modify different portions of your installation project.



For example, the General Information view is where you set general product and project properties; the Setup Design view enables you to edit the features, components, and component data used by your project; the Registry view enables you to modify the registry data installed by your installation program; and the Direct Editor view gives you access to the raw MSI database tables.

It is also assumed you are familiar with some of the wizards available with InstallShield, such as the Release Wizard and Component Wizard.

- **The Release Wizard**, available under the Build menu and also from the Releases view, lets you describe the properties—media type, compression settings, and so forth—of a release, and then builds the specified release image.
- **The Component Wizard**, available by right-clicking a feature in the Setup Design view, lets you create special types of components, such as components for COM servers, fonts, and Windows services.

The InstallShield Help Library contains information about using every view and wizard in the InstallShield environment. The InstallShield Help Library is available when you press F1 with any view selected; you can also select Contents from the Help menu to view the help library.

In addition to the graphical environment, InstallShield provides several tools for modifying and building projects from the command line or an external script. For example, to build a project from the command line, batch file, or other automated process, you can use the executable IsCmdBld.exe. The IsCmdBld executable is located in the System subdirectory of the InstallShield distribution directory.

To rebuild a project, you pass IsCmdBld the project file path, the product configuration name, and the release name that you want to rebuild. A sample command appears as follows:

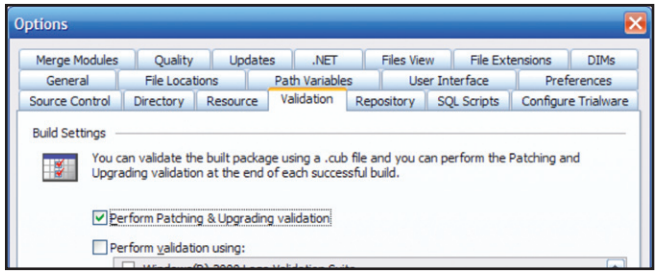
```
iscmdbld -p C:\ProductName.ism -a BuildConfig -r ReleaseName
```

In addition, InstallShield provides an Automation interface, with which you can modify the contents of a project file without using the graphical environment.

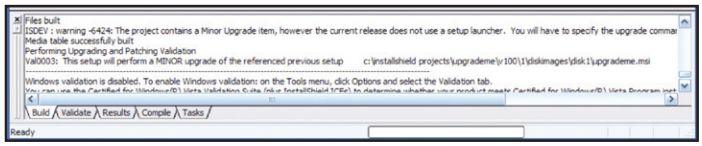
Learn More about InstallShield
If you wish to learn more about the capabilities of InstallShield, please visit the Acresso Web site at www.acresso.com/installshield

Performing Upgrade Validation

By default, InstallShield will perform upgrade validation each time you build a release image. You can change this by pulling down the Tools menu, selecting Options, activating the Validation tab, and selecting the desired behavior.



When you perform a build, the upgrade validation results are displayed in the Build tab of the output window at the bottom of the InstallShield environment. Any messages displayed will contain a validation rule number and a description of the project elements involved in the rule.

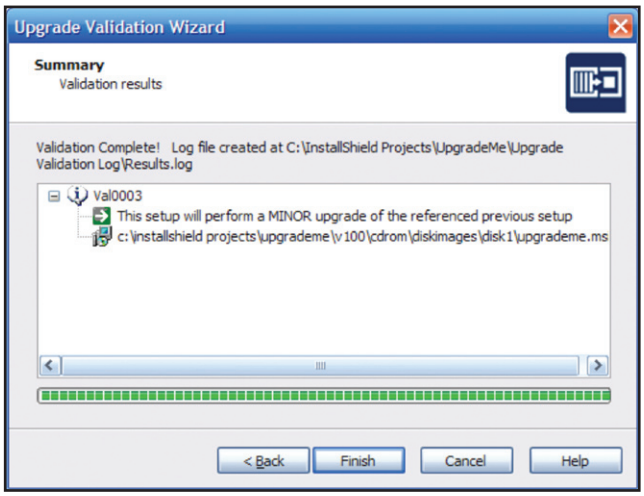


Any validation warnings or errors will also be displayed in the Tasks tab of the output window.

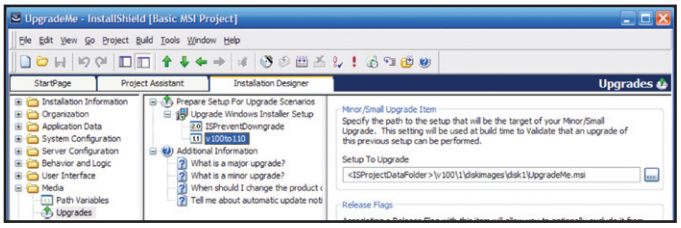
You can click the error number in the Error Code column to obtain more information about that error from the InstallShield technical-support Web site.

If you build a project from the command line or a batch file using Iscmdbld.exe, the update validation results will be displayed at the command prompt and written to the build log file.

You can also use the Upgrade Validation Wizard, in which you browse for your latest and previous product versions, and then run upgrade validation on the two versions. To launch this wizard pull down the Build menu, select Validate, and then select Upgrade Validation Wizard. After you browse for the MSI databases being validated, the wizard displays information, warning, and error messages in the Summary panel, pictured in the following figure.



Similarly, you can right-click an upgrade item in the Upgrades view and select Validate Item to carry out validation on that item.when installed to directories with spaces.



Validation Rules

Many of the InstallShield validation rules apply only to certain types of updates. The categories of upgrade validation rules are the following:

- All types of upgrades: Val003, Val005
- Minor upgrades: Val001, Val002, Val004, Val006, Val007, Val009, Val012
- Major upgrades: Val008, Val013, Val014
- Patches only: Val011, Val015

The following are the built-in upgrade validation rules.

Val001: Files removed from a later product version should have corresponding RemoveFile records. If a file that existed in the earlier version of your project is no longer present in the later version, the minor upgrade process will not remove the file unless you add a corresponding record to the RemoveFile table.

To access the RemoveFile table in InstallShield you can use the Direct Editor view. In the Direct Editor you can select the RemoveFile table, and add records corresponding to files you want to delete.

See also Val002, Val009.

Val002: For minor upgrades, this validation rule displays a note for RemoveFile records that apply only on installation.

The InstallMode field of the RemoveFile table takes a numeric value, specifying that the file(s) listed in the record are to be removed when the associated component is installed, uninstalled, or both. In a minor-upgrade situation, if a RemoveFile record uses the remove-only-during-install flag (value 1), the file listed in the record will not be removed if the component associated with the RemoveFile record is not installed.

This validation rule displays an informational message, not a warning. If you do not intend to update this component, or if you are certain the component will be reinstalled during your minor upgrade, you can safely ignore the message. Alternatively, you can set the InstallMode value to 3, in which case the file might not be removed during the upgrade, but it will be removed when the product is removed.

See also Val001.

Val003: This rule displays various informational messages, such as the type of upgrade to be performed. Displays an error if any of the following are true:

- The package code has not changed between the packages.
- The ProductVersion values of the two versions are identical in the first three fields.
- A major upgrade is being performed, but the wrong upgrade code is in the Upgrade table.

In addition, this rule displays a warning if ProductVersion has not changed at all between the versions.

By default, InstallShield creates a new package code GUID every time you perform a build. If you are creating an update package by other means, you must change the package code for each new release (along with other codes, depending on the type of update).

Val004: For a minor upgrade, displays an error if a change to an existing component's files will prevent the component from being updated.

For versioned files, Windows Installer tests only the component's key file when determining if a component should be installed. The default REINSTALLMODE value of "omus" causes a file to install only over an older version, but not a newer or equal version. A REINSTALLMODE value that includes the "e" or "a" flag will cause an equal file version to install over an existing file. (A drawback to changing the value of REINSTALLMODE this way, however, includes causing unnecessary prompts for the original installation source during the application of a patch. In addition, the REINSTALLMODE value applies to all features being reinstalled, which in the case of the "a" flag can cause older files to overwrite newer ones.)

Val005: For all types of updates, displays a warning if any features have a default install level of 0 (zero), or can conditionally be set to 0.

A feature's install level is a numeric value associated with that feature. At run time, if a feature's install level is strictly greater than the value of the built-in property INSTALLLEVEL, the feature will not be selected for installation. A special case—the case tested by this validation rule—is that a feature install level of zero will deselect the feature and hide it from the feature-selection dialog box.

The issue is that a feature must be installable in order for updated components to be installed. To fix warnings displayed by this validation rule, you should ensure the feature is conditioned to be active during maintenance and update installations.

Val006: This rule displays an error message if a minor upgrade is being performed when a major upgrade should be, because a component or feature was removed from the product tree. A major upgrade is required if a component or feature has been removed from the product tree.

Val007: Ensures the MSI file name has not changed for a minor upgrade package. The MSI file name cannot change between versions in a minor upgrade. Attempting to perform a minor upgrade when the MSI file name has changed can lead to MSI run-time error 1316.

Val008: Validates Upgrade table "action properties" and version data. Major upgrade settings—such as the range of product versions to update on a user's system, and a public action property used to store update-related data at run time—are stored in the Upgrade table of your MSI database. This rule performs the following validation:

- Ensures the action property is included in the value of the SecureCustomProperties property.
- Ensures the action property is a public property, with an all-uppercase name.
- Ensures the action property is not defined in the Property table.
- Displays a warning if an action property is repeated in multiple records.
- Ensures the minimum version is less than the maximum version.

TIP: See also Microsoft validation rule ICE61. ICE61 tests for the same conditions as Val008, with an additional test to ensure a major upgrade package will not attempt to update itself.

If you used the Upgrades view of InstallShield, most of these settings will be enforced automatically.

Val009: Just as files removed from a minor upgrade package require records in the RemoveFile table (tested in Val001), registry data removed from a component must have corresponding data in the RemoveRegistry table. For a minor upgrade, this rule displays a warning or error message if registry data has been removed from a component, but there is missing or inappropriate information from the RemoveRegistry table.

See also Val001.

Val011: For a patch, this rule checks for schema compatibility. Windows Installer does not support patches if the before and after databases have different schemas, such as from MSI 1.2 to MSI 2.0 or from the schema of a "normal" installer to a "large" installer (which alters a data type used by fields in the File, Media, and Patch tables). If the database schema has changed between the earlier and later versions, you will need to package your update as a full minor upgrade or major upgrade package.

Val012: Displays an error message if a minor upgrade package contains new root-level features, or contains new subfeatures that are missing the required settings. A minor upgrade requires any new feature to be a subfeature of an existing feature; and the new feature must be set to be required and to follow its parent feature. If you want, you can make the new subfeature invisible to the end user.

Val013: For a major upgrade, warns about use of the Remove field of the Upgrade table. One field in the Upgrade table is the Remove field, which can optionally contain a list of features to remove. Use of this field will cause only those features listed to be removed, leaving behind resources.

The default and usually most appropriate approach is to leave the Remove field of an Upgrade record empty, which indicates to remove all the features of the earlier version.

Val014: This rule checks for removed components in certain major upgrades. The RemoveExistingProducts action can be scheduled in different locations, to modify the major upgrade behavior. This rule applies to the "Install setup then remove unneeded files" setting, and not the "Completely uninstall old setup..." setting.

TIP: See also Windows Installer validation rule ICE63. ICE63 tests for valid placement of the RemoveExistingProducts action, and displays a warning or error message if the action is in an invalid location. For more information, see the MSI Help Library topics "ICE63" and "RemoveExistingProducts Action".

Val015: This rule warns about package changes that will prevent you from creating an uninstallable patch.

Summary

This white paper gave you an overview of the upgrade validation performed by InstallShield. It supports different validation rules for upgrades and patches, which can assist you in identifying and avoiding potential problems. This update validation can help you enforce design guidelines while you build your update packages.

How Acresso Professional Services Can Help

Knowing about validation rules is different to knowing how to deal with them, which ones to fix, and why. The consequences to your packages and repackaging infrastructure are also something that our experienced consultants can assist with. Find out more at www.acresso.com/services/consulting/software-installations.htm.

Powering the business of software

Acresso Software Inc.
1000 E. Woodfield Road,
Suite 400
Schaumburg, IL 60173 USA

Schaumburg (Global Headquarters),
Santa Clara:
+1 800-809-5659

United Kingdom (Europe,
Middle East Headquarters):
+44 870-871-1111
+44 870-873-6300

Japan (Asia,
Pacific Headquarters):
+81 3-4360-8291

Australia:
+61 2 99-8-22-178

www.acresso.com

IS_WP_Validation_Apr09