

MSI Sequences and Standard Actions: An Introduction

by Robert Dickau
Principal Technical Training Writer, Acreesso Software



MSI Sequences and Standard Actions: An Introduction

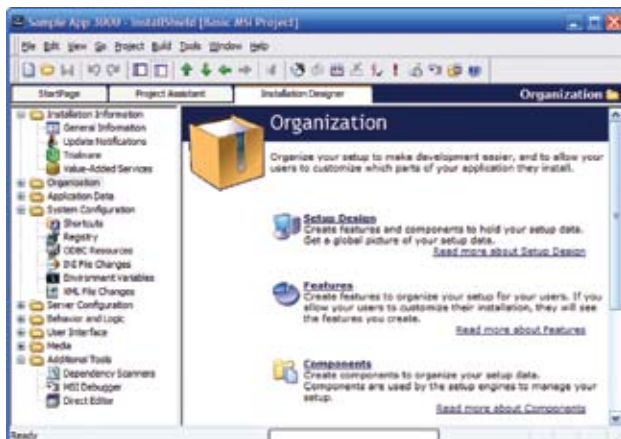
Introduction

Instead of using an explicit script, Windows Installer (MSI) uses *sequences* to specify the order in which operations are performed and the conditions under which they are performed. This white paper introduces Windows Installer sequences, and describes some of the built-in actions that make up a typical installation program.

It also highlights how InstallShield® from Aceso Software makes it easier to work with sequences and standard actions.

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.

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

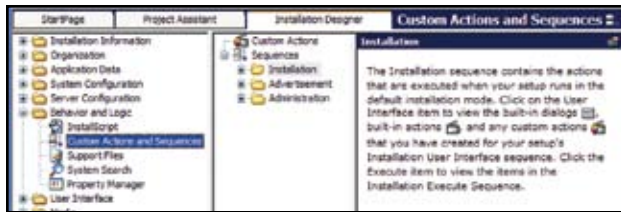
If you wish to learn more about the capabilities of InstallShield, please visit the Aceso Web site at www.aceso.com/installshield.

Top-Level Sequences

Windows Installer supports three types of installation, called a normal installation, administrative installation, and advertised installation.

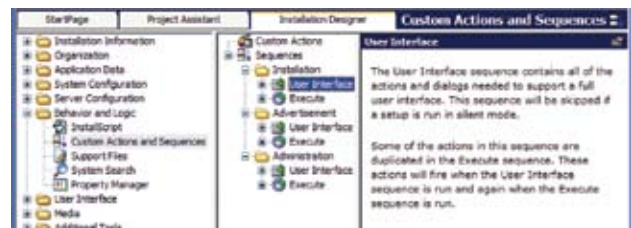
- A normal installation is the type performed when the user double-clicks an MSI database file, or runs `msiexec.exe` with the `/i` switch (or the `/package` switch, starting with Windows Installer 3.0). This white paper focuses on normal installations.
- An administrative installation simply creates an uncompressed version of the installation image in a specified target directory, usually a network location. An administrator performs an administrative installation by launching `msiexec.exe` with the `/a` switch. An administrative installation does not register the product with the target system, in the sense that an administrative installation creates no shortcuts, no registry data, no uninstallation entry, and so forth.
- An advertised installation installs various application “entry points”—shortcuts, file-extension information, and COM information—but does not install any application data (files, registry data, and the like) until the user invokes one of these entry points. Advertised installations are useful for saving disk space on the target system until the product or specific features of it are needed. A user can advertise an installation by passing `msiexec.exe` the `/jm` or `/ju` switch, indicating per-machine and per-user advertisement, respectively.

Each of these installation types is represented by a *top-level sequence*. You can inspect and manipulate these top-level sequences in the Custom Actions and Sequences view of the InstallShield environment, in the “Behavior and Logic” view group.



User Interface and Execute Subsequences

Each of the top-level sequences is divided into two subsequences, called the User Interface (UI) sequence and the Execute sequence. For a normal installation, these sequences are represented by the MSI database tables `InstallUISequence` and `InstallExecuteSequence`. (The tables used for the advertisement and administration sequences are called `AdvUISequence`, `AdvExecuteSequence`, `AdminUISequence`, and `AdminExecuteSequence`.) In the InstallShield environment, each of the subsequences is displayed in a separate tree under the corresponding top-level sequence.



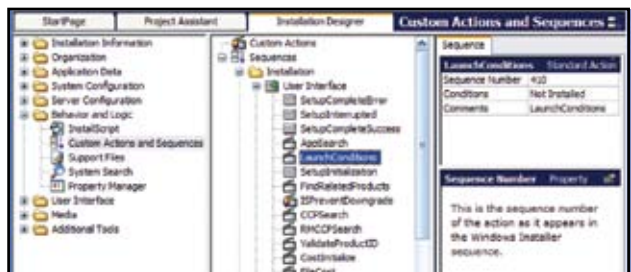
In general, the User Interface sequence shows dialog boxes and queries the user and the target system, but does not perform any system changes. On the other hand, the Execute sequence performs system changes, but does not display any user-interface elements.

A normal installation runs both the User Interface sequence and the Execute sequence. A silent installation performs only the Execute sequence, meaning that any actions in the User Interface will be skipped for a silent installation. Reduced-UI and basic-UI installations, such as an uninstallation launched from the Remove button in the Add or Remove Programs panel, also skip the actions in the User Interface sequence. Similarly, launching an installer in test mode from the InstallShield environment (by clicking the Test toolbar button or pressing **Ctrl+T**) effectively runs only the User Interface sequence.

The User Interface sequence and the Execute sequence run in different processes. The two processes communicate using properties. Only the values of public properties—those with an all-uppercase name, such as `MYCUSTOMPROPERTY`—are passed from the User Interface sequence to the Execute sequence. (Such properties should also be listed in the `SecureCustomProperties` property.) Values of private properties—those with a name that contains a lowercase letter, such as `Privileged`—will be reset to their defaults when execution switches from the User Interface sequence to the Execute sequence.

Introduction to Actions

Sequences are ordered lists of *actions*, where an action corresponds to one operation performed by the installation. If you expand a sequence icon in the Sequences tree, you can view the individual actions contained in that sequence. The following figure shows some of the actions appearing in the User Interface sequence of a typical installation project.



Different types of actions are represented by different icons. Dialog actions are represented by the following icon (dialog actions can occur only in the User Interface sequence):



Standard actions—built-in actions provided by the Windows Installer service—are represented by the following icon:



Custom actions are represented by the following icon:



By default, every new InstallShield project contains several custom actions.

Each action is represented by a record in one of the sequence tables (InstallUISequence, InstallExecuteSequence, and so forth). Each record in the sequence tables contains Action, Condition, and Sequence fields. In general, actions run in order of ascending Sequence number. (Rollback actions run in the opposite order.) The Custom Actions and Sequences view of the InstallShield environment displays actions in order of increasing sequence number. To move an action to a different point in a sequence or from one sequence to another, you can drag it from the old location and drop it onto the new location.

NOTE: One case where an action can be called without appearing in the sequence tables is an action launched with a DoAction event associated with a dialog box control, such as a button. Similarly, some types of custom actions can call other actions using the MsiDoAction function or the Session.DoAction method.

You can also view actions in the Direct Editor view, selecting the desired table and sorting the records by sequence number.



In the Sequences view (or the Direct Editor view), you can also modify the condition associated with an action. For example, to specify that an action should run only on Windows 2000 or later, you can give the action the condition **VersionNT>=500**, or to specify that an action should run only when the installer has been launched or deployed by an administrator, you can use the condition **Privileged**. If an action has no condition associated with it, it will always run.

To insert an action into the sequences, you can right-click the action that should precede your action and select Insert. In the Insert Action panel that appears, you can select the type of action you want to insert (standard action, custom action, merge module action, and so forth), select the specific action, and specify an optional condition for the action.

You can also place actions in the sequences using the Custom Actions tree in the Custom Actions and Sequences view.

Exceptions to the rule that Windows Installer executes actions in order of increasing sequence number are the various “setup complete” dialog boxes, which have negative sequence numbers in the User Interface sequence.



The negative sequence numbers that have special significance to Windows Installer are the following:

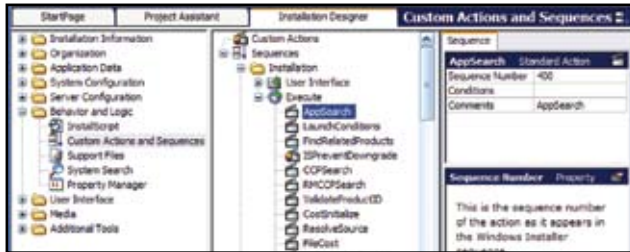
- **-1 for the SetupCompleteSuccess** dialog box, displayed at the end of a successful installation.
- **-2 for the SetupInterrupted** dialog box, displayed if the user cancels the installation.
- **-3 for the SetupCompleteError** dialog box, displayed if the installation fails because a fatal error occurred.

Windows Installer provides several dozen standard actions that carry out an installation. Some of the categories of standard actions are:

- Disk-costing (or file-costing) actions, which determine whether the target system has enough disk space to install the application. Actions in this category include CostInitialize, FileCost, and CostFinalize.
- Actions that handle common system changes, which include manipulation of files, directories, registry data, environment variables, INI file data, and so forth. Actions in this category include InstallFiles, CreateFolders, WriteRegistryValues, WriteEnvironmentStrings, and WriteIniValues.
- Script-generation actions, which handle creation of the internal script that Windows Installer uses to carry out an installation. Actions in this category include the InstallInitialize and InstallFinalize actions.

InstallShield provides many custom actions to enhance Windows Installer functionality, enabling support for executing SQL scripts, searching and modifying XML files, creating Web sites and virtual directories for IIS, and more.

As described earlier, the Execute sequence contains the actions that make changes to the target system.



For standard actions, Windows Installer typically waits for one action to complete before continuing in the sequences to the next action. One consequence of this behavior is that Windows Installer does not support performing part of a standard action, running custom code, and then completing the standard action.

For example, an installation cannot install only some files, launch a custom action, and then complete the file-transfer action. Instead, the standard InstallFiles action occurs as a single operation, and a custom action must be called either before or after InstallFiles, but not during InstallFiles.

Summary

This white paper introduces Windows Installer sequences, and describes some of the built-in actions that make up a typical installation program. It also highlights how InstallShield® from Aceso Software makes it easier to work with sequences and standard actions.

Begin a Free Evaluation of InstallShield

You can download a free trial version of InstallShield from the Aceso Software Web site at:
www.aceso.com/installshield/eval

Want to learn more best practices for building quality installations? Join an InstallShield training class – visit
www.aceso.com/training for available classes.



Acesso Software Inc.
900 National Parkway, Suite 125
Schaumburg, IL 60173
USA

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

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

Japan (Asia, Pacific Headquarters):
+81 3-5774-6253

www.acesso.com