

Working with MSI Features and Components

Features represent the end user's view of your MSI installation program, and components represent your view of the installation. Part of your responsibility as the installation designer is to decide how many separately installable pieces of the installation should be presented to the user. Each of these pieces should be a feature or subfeature. For more tips, download an InstallShield® white paper at www.flexerasoftware.com/resources/whitepapers.htm

When designing features and components, you should consider the following points:

- All of the file contents of a component must be installed to the same directory. If application files need to be installed in multiple directories, you must create at least one component for each destination.
- A component is the lowest level to which a condition can be attached. There is no method for installing only some of the files in a component, or for installing the files but not registry data for a component. If you have data that need to be installed under different conditions—such as different target operating systems or languages—the data must be separated into different components.
- No resource—file, registry key, shortcut, and so forth—should be placed in more than one component, even across different products and organizations. (Placing the same resource in multiple components breaks Windows Installer reference counting.) Instead, if a resource is required by multiple pieces of the application, you can share entire components among multiple features, or use merge modules to share components among multiple products.
- For the most effective file transfer, a component should contain at most one executable or DLL—ideally, a versioned file—and this file should be marked as the key file of its component.
- Because component and feature names can occasionally be returned in Windows Installer log files, for the sake of maintainability you should give components and features descriptive internal names. In the case of components, one common practice is to name the component after its key path (usually its key file).
- When there are identical properties for both components and features, the component property is the setting used. For example, both components and features have a Destination property. The feature's Destination property is the one displayed to the end user in the Custom Setup dialog box, while the component's Destination property is the one actually used. In the common special case of the values being the same public property (as in INSTALLDIR), the destination selected in the Custom Setup dialog box is passed along to the component.

Begin a Free Evaluation of InstallShield

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

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