

1, 2, ... command (Window menu)

InstallShield displays a list of currently open windows at the bottom of the Window menu. A check mark appears in front of the name of the active window. Select a window from this list to make it the active window.

{button ,PI("",`feedback')} Feedback

About InstallShield command (Help menu)

This command opens the About InstallShield dialog box. The About InstallShield dialog box displays the copyright notice and version number of your copy of InstallShield, and gives you access to version information for individual InstallShield product files.

{button ,AL(`About InstallShield dialog box',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Readme command (Help menu)

Launches your Web browser and opens Readme.htm, which provides introductory information such as system requirements, setup instructions, and an overview of InstallShield's documentation.

{button ,PI("",`feedback')} Feedback

Product News command (Help menu)

Launches your Web browser and points it to the InstallShield Web page providing links to InstallShield newsgroups and up-to-date product information.

{button ,PI("",`feedback')} Feedback

Knowledge Base command (Help menu)

Launches your Web browser and points it to the InstallShield Web page that provides access to the InstallShield Knowledge Base. The Knowledge Base contains answers to many commonly asked questions and includes new information about InstallShield that may not appear in the documentation. You can use the search engine provided on the Knowledge Base to search articles by phrases, numbers, platforms, and version.

{button ,PI("",`feedback')} Feedback

Online Support command (Help menu)

Launches your Web browser and points it to the InstallShield Web page that lets you request product support or learn about InstallShield support programs.

{button ,PI("",`feedback')} Feedback

Send Feedback command (Help menu)

Launches your Web browser and points it to the InstallShield Web page that lets you send comments on InstallShield.

{button ,PI("",`feedback')} Feedback



Register InstallShield command (Help menu)

Launches your Web browser and points it to the InstallShield Web page that lets you register your copy of InstallShield.

{button ,PI("",`feedback')} Feedback

InstallShield Home Page command (Help menu)

Launches your Web browser and points it to InstallShield's home page.

Shortcut

Toolbar: 

{button ,PI("",`feedback')} Feedback

About InstallShield dialog box

Opens when you select About InstallShield from the Help menu. Displays the copyright notice and version number of your copy of InstallShield, and gives you access to version information for individual InstallShield product files.

Controls

About InstallShield Professional (hypertext link)

Launches your Web browser and points it to the InstallShield home page.

OK (button)

Closes the About InstallShield dialog box.

Files (button)

Opens the Product Files dialog box, which lists filenames and version numbers for all files provided with your copy of InstallShield.

{button ,AL(`Product Files dialog box',0,`,`')} See also

{button ,PI(`",`feedback')} Feedback

Add File Group dialog box

Opens when you click the Add button in the Included File Groups page of the Component Properties sheet. Allows you to associate file groups with the active component (the component selected in the Components pane).

Controls

File Group Name (list box)

Contains the names of all defined file groups.

File Groups (edit box)

Displays the name of a file group selected from the File Group Name list box, or allows you to enter the name of a file group not in the list.



Entering a name of a nonexistent file group does not create that file group. You must create the file group in the File Groups pane.

{button ,AL(`File Groups page (Component Properties sheet);Associate file groups with a component I',0,'')}

See also

{button ,PI("",`feedback')} Feedback

Add Language dialog box

Opens when you click the Add button in the Language page of the Project Properties sheet. Lets you add a language to those supported by the active project (the project selected in the New dialog box).

Controls

Show only available languages (check box)

If checked, the only languages shown are those for which you have purchased InstallShield resources.

Show sublanguages (check box)

If checked, shows variants of base languages, for example, Norwegian (Bokmal) and Norwegian (Nynorsk).

{button ,AL(`Language page (Setup Properties sheet)',0,`,`)} See also

{button ,PI(`",`feedback')} Feedback

Add Platform dialog box

Opens when you click the Add button in the Operating Systems page of the Project Properties sheet. Lets you add an operating system to those supported by the active project (the project selected in the New dialog box).

{button ,PI("",`OS_limits')} Visual C++ edition limitation.



16-bit platforms such as Windows 3.1 can't read long filenames. If the platforms targeted by your setup include a 16-bit platform, don't use long filenames for your application files.

Controls

Show only available platforms (check box)

If checked, the only operating systems shown are those for which you have purchased InstallShield compatibility.

{button ,AL(`Operating Systems page (Setup Properties sheet)',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Advanced Build Type property sheet

Opens when you click the Advanced button in the Media Build Wizard's Build Type panel. Contains the following property pages:

Date/Time

Lets you set the date and time stamps of files.

Disk Information

Lets you "reserve" (leave empty) space on your distribution disks.

Build Location

Lets you specify the location for your built disk images.

Setup

not in visual c++ edition

Lets you modify the startup message, or offer your end user a choice of languages.

Password

Lets you specify a password for the entire setup.

{button ,AL(`Build Type dialog`)} See also

{button ,PI("",`feedback`)} Feedback

InstallShield for Visual C++ does not allow you to modify the startup message.

Bookmark dialog box

Opens when you press Alt+F2 and your cursor is in a script editor window. Lets you place a named bookmark at a particular line and column position in the script file or go to an already existing bookmark in the file.

Controls

Name (combo box)

Enter a name for a new bookmark in the edit box, or select the name of an already existing bookmark from the list box.

Add (button)

Adds a bookmark, with the name in the Name edit box, at the position of the cursor. (The line number of the cursor position is displayed at the bottom of the Bookmark dialog box.)

Close (button)

Closes the dialog box.

Delete (button)

Deletes the bookmark whose name appears in the Name edit box.

Go To (button)

Places the cursor at the position of the bookmark whose name appears in the Name edit box.

{button ,PI("", `feedback')} [Feedback](#)

Build tab (Output window)

If you select the Build tab, the output window displays progress and error messages from the file media library builder.

{button ,PI("",`feedback`)} Feedback

Build Location page (Advanced Build Type property sheet)

Lets you specify the location for your disk images folders and files and other build-related output such as build log files.



InstallShield does not let you specify a root folder (for example, C:\) as your build location. This is because when you delete the project and its data and media files using the Delete Project dialog box, all files and folders under the build location are deleted.

Controls

Location (edit box)

Displays the folder in which your disk images folders will be stored.

Browse (button)

Opens the Choose Directory dialog box, in which you can select a folder.

{button ,AL(`Advanced Build Type property sheet;Delete Project dialog box')} [See also](#)

{button ,PI("", `feedback')} [Feedback](#)

Build Type panel (Media Build Wizard)

Lets you specify the build type—Full or Quick—and advanced build properties.

Controls

Full Build (option button)

Creates disk image files that contain actual compressed copies of your application files. (Previously created disk image files are overwritten.) Use this option to build your distribution media.

Quick Build (option button)

Creates disk image files that contain links to your actual application files; that is, the disk image files redirect the InstallShield setup engine to your application files. As a result, the setup runs only on the system on which the application files are stored.



You can use this option to test your setup; the disk image files are much smaller and faster to build than disk image files containing actual files.

Review Report Before Build (check box)

not in visual c++ edition

If checked, InstallShield displays the Media Library Report dialog box before building the file media library.

Advanced (button)

Opens the Advanced Build Type dialog box.



Clicking the Cancel button results in no build being created, and no existing build being overwritten.

{button ,AL(` Advanced Build Type property sheet;Name:Media Library Report dialog box',0,'')} See also

{button ,PI("",`feedback')} Feedback

Build menu commands

The Build menu offers the following commands:

Compile

Compiles the project script file Setup.rul.



The compiler compiles only Setup.rul. Other script files are compiled only if they are included in Setup.rul with a #include statement, and only when Setup.rul is compiled.

Run Setup

Compiles and runs the setup.

Debug Setup

Compiles and runs the setup in debug mode.

Build Media

Launches the Media Build Wizard.

Send Media To

Launches the Send Media Wizard.

Settings

Opens the Setup Settings dialog box.

{button ,PI("",`feedback')} Feedback

Building Media panel (Media Build Wizard)

Displays a continuously updated report on the status of the build process.

Controls

Cancel (button)

Abort the build process.

Finish (button)

Exit the Media Build Wizard after the build process has ended.



Clicking the Cancel button results in no build being created, and no existing build being overwritten.

{button ,PI("", `feedback')} [Feedback](#)

Compile command (Build menu)

Compiles your project's script. Messages are displayed in the Output window's Compile tab.



The compiler compiles only Setup.rul. Other script files are compiled only if they are included in Setup.rul with a #include statement, and only when Setup.rul is compiled.

Shortcuts

Toolbar: 

Keys: Ctrl+F7

{button ,AL(`Output window;Compiling the setup script l',0,'`')} See also

{button ,PI(`",`feedback')} Feedback

Run Setup command (Build menu)

Runs the setup that is on the project's most recently built disk images.

Shortcuts

Toolbar: 

Keys: Ctrl+F5

{button ,PI("",`feedback')} Feedback

Debug Setup command (Build menu)

Runs the setup that is on the project's most recently built disk images from within the Visual Debugger. In the Visual Debugger you can examine each line of your script as it executes.

Shortcuts

Toolbar: 

Keys: F5

{button ,AL(`debug |,0,`,`)} See also

{button ,PI(`",`feedback')} Feedback

Media Build Wizard command (Build menu)

Launches the Media Build Wizard, which builds disk images for your setup.

Shortcut

Toolbar: 

{button ,AL(`Media Build Wizard',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Send Media To command (Build menu)

Launches the Send Media Wizard, which helps you copy your disk images to disks in a disk drive or another folder.

{button ,AL(`Send Media Wizard',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Settings command (Build menu)

Opens the Setup Settings dialog box, in which you can make settings relevant to the compiling and debugging of your script and the running of your setup.

Shortcut

Keys: Alt+F7

{button ,AL(`Setup Settings dialog box',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Setup Settings dialog box

Opens when you select Settings from the Build menu. Lets you make settings relevant to the compiling and debugging of your script and the running of your setup. Contains the following tabs:

[Compile](#)

[Run/Debug](#)

{button ,PI("",`feedback`)} Feedback

Run/Debug tab (Setup Settings dialog box)

The Setup Settings dialog box opens when you select Settings from the Build menu. The Run/Debug tab lets you make settings relevant to the running and debugging of your setup.

Controls

Program Arguments (edit box)

Text placed in this box is written to the system variable CMDLINE when you run the setup; this variable can be accessed in the script.

Test Mode (check box)

If this box is checked, your actual files are not copied to their target locations when you run the setup. Instead, 0-kilobyte files with the same names are copied to the target locations (except for shared files, which are copied normally whether or not this box is checked).

Launch DLL Debugger (check box)

{button ,PI("",`dll_limit`)} Visual C++ edition limitation.

If this box is checked, the debugger specified in the Debugger Filename edit box runs when a DLL is called during setup debugging.

Debugger Filename (edit box)

Specify the debugger that runs when a DLL is called during setup debugging. (Disabled if the Launch DLL Debugger check box is unchecked.)

Generate MIF File (check box)

If checked, InstallShield generates a Management Information Format (.mif) file automatically at the end of the setup.

MIF Filename (edit box)

Specifies the name of the .mif file generated by InstallShield. If left empty, the .mif file is named Status.mif.

Product Serial Number (edit box)

If a serial number is specified, InstallShield places it in the .mif file it generates.

{button ,PI("",`feedback`)} Feedback

InstallShield for Visual C++ does not allow prototyping of DLL functions; it offers just the limited CallDLLFx function.

Compile tab (Setup Settings dialog box)

The Setup Settings dialog box opens when you select Settings from the Build menu. The Compile tab lets you make settings relevant to the compiling of your script.



The compiler compiles only Setup.rul. Other script files are compiled only if they are included in Setup.rul with a #include statement, and only when Setup.rul is compiled.

Controls

Warning Level (list box)

Specifies what kinds of warnings are displayed in the output window's Compile tab.

Level 1 Displays any system warning message that InstallShield is unable to handle.

Level 2 Displays Level 1 messages, plus a message if string length exceeds the limit.

Level 3 (Default) Displays all warning messages.

None Displays no warning messages.

Warnings as errors (check box)

If this box is checked, warnings are treated as errors, that is, the setup is not run if there were any warnings during the compile.

Reset (button)

Sets all controls to their default values.

CRC Generation (check box)

If this box is checked, cyclic redundancy checking is performed on the compiled script.

Max Warnings (edit box)

Specifies the maximum number of warning messages displayed.

Max Errors (edit box)

Specifies the maximum number of error messages displayed.

Preprocessor Defines (edit box)

Specify any preprocessor variable definitions. Use the following format, with no spaces before or after equals signs or commas:

```
MYVARIABLE1=123,MYVARIABLE2=456
```

Such variables can be tested in the script by #if and #ifdef statements that control the flow of the script.



Entering the name of a preprocessor variable in this edit box defines the variable. For example if you enter MYVARIABLE in this edit box, the script commands in the following #ifdef loop are executed:

```
#ifdef MYVARIABLE  
    // Commands  
#endif
```



After you add or change a preprocessor variable definition in this edit box, you must compile your setup project for the addition or change to take effect. Your project is compiled when you do any one of the following:

- n Click the Compile Setup button  on the toolbar.
- n From the Build menu, select Compile.
- n Press Ctrl+F7.
- n Run your setup from within the IDE after making a change to the setup script.

- n Run the Media Build Wizard when the “Compile before build” check box is checked (see below).

Compile before build (check box)

If this box is checked, the setup project script is compiled whenever the Media Build Wizard runs.

{button ,AL(`#ifdef and #ifndef I',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Cascade command (Window menu)

Arranges multiple opened windows in an overlapped fashion.

{button ,PI("",`feedback')} Feedback

Choose Dialogs panel (Project Wizard)

Lets you select commonly used dialog boxes you want in your setup. InstallShield automatically places function calls for the selected dialog boxes in your script file.



The dialog boxes listed in the list box are only the most commonly used dialog boxes. InstallShield offers many more dialog boxes, which you can include in your setup by using function calls in your script.

Controls

Dialogs (list box)

Lists commonly-used dialog boxes. Check a dialog box's check box to automatically include it in your setup script.

Preview (button)

Opens an image of the dialog box selected in the Dialogs list box, showing how the dialog box appears to your end user.

{button ,AL(`Setup Wizard;Built-in dialog box functions I',0,'')} See also

{button ,PI("",`feedback')} Feedback

Choose Target Platforms panel (Project Wizard)

Lets you select the operating system(s) on which you want your setup to run.

{button ,PI("",`OS_limits')} Visual C++ edition limitation.



16-bit platforms such as Windows 3.1 can't read long filenames. If the platforms targeted by your setup include a 16-bit platform, don't use long filenames for your application files.

Controls

Operating System (list box)

Displays one of the following:

- n All available operating systems (when "Show only available platforms" check box is checked)
- n All operating systems supported by InstallShield, with unavailable operating systems disabled (when check box is unchecked).

Select the operating systems your setup will target. To select multiple operating systems, hold down the Shift or Ctrl key and click with the mouse.

Show only available platforms (check box)

Controls operating systems displayed in Operating System list box (see above).



Available operating systems are those for which you have purchased setup availability from InstallShield Corporation.

{button ,PI("",`feedback')} Feedback

Close command (Control menu)

Closes the active window, or closes InstallShield if selected from InstallShield's control menu.

Shortcuts

Mouse: Click the close icon  on the title bar.

Keys: Ctrl+F4

Comment page (Component Properties sheet)

Lets you add, view, or modify comment text associated with the active component (the component selected in the Components pane). This text is not included in the final file media library.

Controls

Comment (edit box)

Displays the comment text associated with the active component.

{button ,PI("",`feedback')} Feedback

Comment page (File Group Properties sheet)

Lets you add, view, or modify comment text associated with the active file group (the file group selected in the File Groups pane). This text is not included in the final file media library.

Controls

Comment (edit box)

Displays the comment text associated with the active file group.

{button ,PI("", `feedback')} Feedback

Components pane (project workspace)

Displays a tree control with components and subcomponents. Double-clicking a component brings up the Component Properties window, which displays the properties list of the active component. Double-clicking a property in the properties window brings up a property page in which you can review or modify that property.

Components can be reorganized by dragging and dropping them; for instructions, click [here](#).



Components are installed in the order in which they appear in the Components pane.



When naming components, keep in mind that end user component dialog boxes can display component names up to a maximum of 45 characters, except for the SdComponentMult dialog box, which can display component names up to a maximum of 20 characters.

{button ,PI("",`comp_limits`)} Visual C++ edition limitation.

{button ,AL(`Components property sheet;Components pane popup menu;SdComponentMult I',0,`,``)} See also

{button ,PI("",`feedback`)} Feedback

Components pane popup menu

New Component

Creates a component in the selected component set, or a subcomponent of the selected component.

Delete

Deletes the selected component.

Rename

Renames the selected component.

Properties

Opens the property sheet of the selected component.

{button ,AL(`Components property sheet;Components pane',0,',';')} See also

{button ,PI("",`feedback')} Feedback

Components property sheet

The Components property sheet contains the following property pages:

Description

Component description shown to your end user.

Status Text

Text shown in progress indicator during installation of component.

Display Name

Component name shown to your end user.

Visible

Whether component is visible to your end user.

Overwrite

Options for installing over already installed files.

Destination

Destination folder for component's files.

File Need

Warning to end users who deselect component.

Include in Build

Whether component is included in built disk images.

Password

Password for component.

Encryption

Whether component's files are encrypted.

CD-ROM Folder

Where component's files are placed on CD-ROM.

FTP Location

FTP location associated with component.

HTTP Location

HTTP location associated with component.

Miscellaneous

Text string associated with component (accessible from script).

Comment

Text string associated with component (not accessible from script).

Required Components

Which other components must be installed when active component is installed.

Included File Groups

File groups associated with component.

{button ,PI("", `feedback`)} Feedback

Compressed page (File Group Properties sheet)

Lets you compress the files in the active file group (the file group selected in the File Groups pane) or leave them uncompressed when the files are stored in a cabinet file by the file media library builder.

{button ,PI("", `compress')} Visual C++ edition limitation.



If you check the "Data as files" check box in the Media Build Wizard's Disk Type panel, application files are not stored in a cabinet file and are not compressed, regardless of the option you select in the Compressed property page.

{button ,AL(`Disk Type panel (Media Build Wizard)',0,`,`')} See also

{button ,PI("", `feedback')} Feedback

InstallShield for Visual C++ does not offer enhanced compression; its compression is the same as that of InstallShield3.

Copy command (Edit menu)

Copies selected text from the active script editor window onto the clipboard. This command is unavailable if there is no text currently selected.

Copying text to the clipboard replaces the contents previously stored there.

Shortcuts

Toolbar: 

Keys: Ctrl+C

{button ,PI("",`feedback')} Feedback

Cut command (Edit menu)

Remove the currently selected text from the active script editor window and puts it on the clipboard. This command is unavailable if there is no text currently selected.

Cutting data to the clipboard replaces the contents previously stored there.

Shortcuts

Toolbar: 

Keys: Ctrl+X

{button ,PI("",`feedback')} Feedback

Date/Time tab (Advanced Build Type property sheet)

Lets you automatically set the date and time stamps of your setup files, or set them manually. If your data is stored as uncompressed files, you can also automatically set the date and time stamps of these data files, or set them manually.

Controls

Setup Files: Build (option button)

Sets the date and time stamps on setup files to be the time of the current build.

Setup Files: User Defined (option button)

Lets you specify the date and time stamps on setup files, in the Date and Time edit boxes.

The following controls are disabled unless the User Defined option button is selected.

Setup Files: Date (edit box)

Enter the date stamp for setup files. The format is mm/dd/yyyy, for example, 10/25/1997.

Setup Files: Time (edit box)

Enter the time stamp for setup files. The format is hh:mm:ss, for example, 14:30:00.

The following controls are disabled unless the "Data as files" check box is checked in the Media Build Wizard's Disk Type panel.

Data Files: Original (option button)

Sets the date and time stamps on data files to be the same as those of the data files on your system.

Data Files: User Defined (option button)

Lets you specify the date and time stamps on data files, in the Date and Time edit boxes.

The following controls are disabled unless the User Defined option button is selected.

Data Files: Date (edit box)

Enter the date stamp for data files. The format is mm/dd/yyyy, for example, 10/25/1997.

Data Files: Time (edit box)

Enter the time stamp for data files. The format is hh:mm:ss, for example, 14:30:00.

{button ,AL(`Disk Type panel (Media Build Wizard)',0,'')} See also

{button ,PI("",`feedback')} Feedback

Description page (Component Properties sheet)

Lets you add, view, or modify the text describing the active component (the component selected in the Components pane). Your end users see this description when they select this component in the SdComponentDialog2 or SdComponentMult dialog box.

Controls

Current Text (static text field)

Displays the current text of the component description.

New Text (edit box)

Lets you enter text for the component description.

>> (button)

Opens the String Entry dialog box. In the String Entry dialog box, click the OK button to specify the string resource in the String Identifier list box as the component description text.

{button ,AL(`String Entry dialog box;SdComponentDialog2 I;SdComponentMult I',0,';')} See also

{button ,PI("",`feedback')} Feedback

Disk ID panel (Media Build Wizard)

Appears if you placed any files in the Setup Files pane's Advanced\Other folder. Lets you specify on which disk of a multi-disk distribution each file should be placed.

Controls

Disk ID/File Name (list box)

Click a disk ID entry to edit it.

{button ,AL(` Setup Files pane (project workspace)',0,`,`)} See also

{button ,PI(`",`feedback')} Feedback

Disk Information page (Advanced Build Type property sheet)

Opens when you click the Advanced button in the Media Build Wizard's Build Type panel. Lets you "reserve" (leave empty) specified amounts of space on particular disks.

Controls

Disk ID/Size Reserved (list box)

Displays disks on which space has been reserved, and the amount of space (in kilobytes) reserved.

Add (button)

Opens the Disk Size Reserved dialog box, with Disk ID set to the disk after the last one on which space is reserved.

Modify (button)

Opens the Disk Size Reserved dialog box, with Disk ID set to the disk selected in the Disk ID/Size Reserved list box.

Remove (button)

Removes the selected entry in the Disk ID/Size Reserved list box.

{button ,AL(`Disk Size Reserved dialog box',0,`,`')} See also

{button ,PI(`",`feedback')} Feedback

Disk Size Reserved dialog box

Comes up when you click the Add or Modify buttons in the Disk Information page of the Media Build Wizard's Advanced Properties sheet. Lets you add or modify entries in the Disk Information page.

Controls

Disk ID (edit box)

Enter the number of the disk on which you want to you "reserve" space (that is, leave empty space).

Size Reserved (edit box)

Enter the amount of space (in kilobytes) you want to reserve on the disk specified in the Disk ID edit box.

{button ,AL(`Disk Information page (Advanced Build Type property sheet)',0,`,`)} See also

{button ,PI(`",`feedback')} Feedback

Disk Type panel (Media Build Wizard)

Lets you select a disk type for your build, and properties of CD-ROM builds.

Controls

Type (list box)

Select a disk type. The InstallFromTheWeb disk type optimizes your files for use with the [InstallFromTheWeb Internet setup tool](#).

Custom Size (edit box)

If you select a custom size CD-ROM in the Type list box, you can enter a disk size in kilobytes here. The default value is 650,000 KB.

Data as files (check box)

If you select a CD-ROM disk type (default or custom size) in the Type list box, you can check this check box to place files uncompressed on the CD-ROM. Files associated with a given component are placed on the CD-ROM in the folder specified in the Component Properties sheet's CD-ROM Folder page. If no folder is specified, that component's files are placed on the root of the CD-ROM.

If this check box is unchecked, files will be stored in cabinet files.



You can specify in a file group's Compressed property page whether the file group's files are compressed or uncompressed when stored in a cabinet file.



Clicking the Cancel button results in no build being created, and no existing build being overwritten.

{button ,AL(`Compressed page (File Group Properties sheet);CD-ROM Folder page (Component Properties sheet)',0,'')} See also

{button ,PI(`",`feedback')} Feedback

InstallShield help is attempting to send you to the InstallFromTheWeb Web site (<http://www.installshield.com/iftw>).

Display Name page (Component Properties sheet)

Lets you add, view, or modify text to be displayed in end-user dialog boxes instead of the name of the active component (the component selected in the Components pane).

Controls

Current Text (static text field)

Displays the component's display name. If this edit box is empty, the component name will be displayed in component dialog boxes during the setup.

New Text (edit box)

Lets you enter component display name.

>> (button)

Opens the String Entry dialog box. In the String Entry dialog box, click the OK button to specify the string resource in the String Identifier list box as the component display name.

{button ,AL(`String Entry dialog box',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Edit menu commands

The Edit menu offers the following commands:

Undo

Undoes the last text entry or deletion.

Cut

Removes the selected text from the script editor window and puts it on the Windows clipboard.

Copy

Copies selected text to the Windows clipboard.

Paste

Inserts a copy of the clipboard contents at the insertion point.

Find

Opens the Find dialog box for finding and marking text.

Replace

Opens the Replace dialog box for replacing text.

Go To

Opens the Go To Line dialog box, which moves your cursor to a specified line number.

Properties

Opens the Script Properties page.

{button ,PI("", `feedback')} Feedback

Edit tab (Options dialog box)

The Edit tab contains settings for vertical and horizontal scroll bars, drag-and-drop text editing, and line selection.

Controls

Vertical scroll bar (check box)

If checked, script editor windows have vertical scroll bars; if unchecked, they don't.

Horizontal scroll bar (check box)

If checked, script editor windows have horizontal scroll bars; if unchecked, they don't.

Drag-and-drop text editing (check box)

If checked, you can drag selected text from one position to another in a script editor window. Dropped text is always inserted, rather than overwriting text at the new position.

Selection margin (check box)

If checked, script editor windows have a left margin in which you can click to select an entire line of text.

{button ,PI("",`feedback')} Feedback

Find command (Edit menu)

Opens the Find dialog box, with which you can search for text in the active script editor window. You can match case or match whole words, and you can make your search proceed forwards, backwards or through the entire file. You can also search for specified text and automatically add line markers to all lines containing the found text.

Shortcut

Keys: Ctrl+F

{button ,AL('Find dialog box',0,','')} See also

{button ,PI('','feedback')} Feedback

Find dialog box

Opens when you select Find from the Edit menu. Lets you search for text and add line markers in the active script editor window.

Controls

Find what (combo box)

Enter the text to search for, or select search text from the drop-down list box.

Match whole word only (check box)

If checked, finds only whole words; if unchecked, also searches for your text inside longer words.

Match case (check box)

If checked, only text with the same pattern of upper and lower case letters that you specified in "Find what" is found.

Direction: Up/Down (option button)

Controls the direction of the text search.

Find Next (button)

Finds the next occurrence of the text in "Find what" in the direction specified by the Direction option buttons.

Mark All (button)

Finds all occurrences of the text in "Find what" and places a line marker at the beginning of each line containing found text.

{button ,AL(`Line markers',0,`,`')} See also

{button ,PI(`",`feedback')} Feedback

Replace command (Edit menu)

Opens the Replace dialog box, which lets you replace text in the active script editor window.

Shortcut

Keys: Ctrl+H

{button ,AL('Replace dialog box',0,'')} See also

{button ,PI("",`feedback')} Feedback

Go To command (Edit menu)

Opens the Go To Line dialog box, which lets you go to a particular line number in the active script editor window.

Shortcut

Keys: Ctrl+G

{button ,AL('Go To Line dialog box',0,'')} See also

{button ,PI("",`feedback')} Feedback

Go To Line dialog box

Opens when you select Go To from the Edit menu. Lets you go to a particular line number in the active script editor window.

Controls

Line Number (combo box)

Enter a line number, or select a line number using the scroll buttons on the right side of the box.

OK (button)

Places the cursor at the beginning of the line specified in the Line Number box.

{button ,PI("",`feedback')} Feedback

Properties command (Edit menu)

Opens the property sheet of the active script (the script whose editor window is active).

Shortcut

Keys: Alt+Enter

{button ,AL(`Script Properties sheet',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Script Properties sheet

Displays information about the active script (the script whose editor window is active).

{button ,PI("",`feedback')} Feedback

Encryption page (Component Properties sheet)

Lets you specify whether the files for the active component (the component selected in the Components pane) will be encrypted. Encrypted files can be read only by the InstallShield engine (or by an expert code-breaker).



If you plan to offer end users the option of running your application directly from the distribution media, do not encrypt the relevant files. InstallShield only decrypts files during installation to the end user's hard drive.



Encryption is not the same as password protection. Encrypting a file scrambles it so that if someone opens it in a text editor, they cannot read strings and other data to get confidential information about your application. Password protecting a component prevents it from being installed by someone who does not know the password. You can encrypt a component's files or you can password protect the component—or, for maximum security, you can do both.

If a component's files are encrypted but neither the component nor the setup is password protected, the component can be installed by anyone who obtains your distribution media (or a copy). If a component is password protected but its files are not encrypted, someone can open the files in a text editor and read confidential information—possibly including the component's password.

{button ,AL(`Password page (Component Properties sheet)',0,'')} See also
{button ,PI("",`feedback')} Feedback

Close command (File menu)

Closes the active project. If you have made changes to the project since you last saved it, InstallShield asks whether you want to save the changes before you close the project. If you close a project without saving, you lose all changes made since the last time you saved it.

{button ,PI("",`feedback')} Feedback

Exit command (File menu)

Ends your InstallShield session. You can also use the Close command on the application Control menu. InstallShield prompts you to save projects with unsaved changes.

Shortcuts

Mouse: Double-click the application's Control menu button.

Keys: Alt+F4

{button ,PI("",`feedback')} Feedback

File Groups pane (project workspace)

The File Groups pane displays a tree control with file groups. Double-clicking a file group folder opens the File Group Properties window, which displays the setup properties of the selected file group. Double-clicking a property in the File Group Properties window brings up a dialog box in which you can review or modify that property. Double-clicking a file group folder's Links subfolder opens the File Group Links window, which displays the file group's files.



InstallShield cannot handle project file paths with more than 260 characters (including the file name). The names of the following project elements are incorporated in project paths and file names:

- n the project itself
- n file groups
- n [media names](#)
- n files in the Scripts pane
- n files inserted into the Setup Files pane

To help you keep within the 260-character limit, InstallShield warns you if you attempt to assign a name with more than 50 characters to a project, file group, or media name.



Do not place folders named TopDir or General in the root of a file group; these names are reserved in InstallShield. Placing folders named TopDir or General in the root of a file group can have undesired results, including non-installation of some files in the file group.

{button ,PI("",`fg_limits')} [Visual C++ edition limitation.](#)

{button ,AL(`File Groups pane popup menu;File Groups property sheet;Files window',0,`,`')} [See also](#)

{button ,PI("",`feedback')} [Feedback](#)

InstallShield for Visual C++ does not let you create more than eight file groups.

File Groups pane popup menu

New File Group

Creates a new file group in the selected file group set, or the file group set in which the selected file group resides.

New Folder

Creates a new folder under the selected Links folder.

Insert Files

Opens the Insert File Links to File Group dialog box, which lets you add files to the selected Links folder.

Delete

Deletes the selected file group.

Rename

Renames the selected file group.

Properties

Opens the File Group Properties window for the selected file group.

{button ,AL(`File Group Properties window;File Groups pane',0,`,``)} See also

{button ,PI(`",`feedback')} Feedback

File Groups property sheet

The File Groups property sheet contains the following property pages:

Shared

Whether the file group's files are shared, that is, used by other applications.

Self Registered

Whether the file group's files are self-registering.

Potentially Locked

Whether the versions of the file group's files that are already on the end user's system might be in use during setup.

Compressed

{button ,PI("", `compress')} Visual C++ edition limitation.

Whether the file group's files are compressed on the distribution media.

Operating System(s)

{button ,PI("", `OS_limits')} Visual C++ edition limitation.

Operating system(s) to which the file group's files are specific.

Language(s)

Language(s) to which the file group's files are specific.

FTP Location

FTP location associated with the file group.

HTTP Location

HTTP location associated with the file group.

Miscellaneous

Text string associated with file group (accessible from script).

Comment

Text string associated with file group (not accessible from script).

{button ,PI("", `feedback')} Feedback

File Need page (Component Properties sheet)

Lets you give a warning, or a severe warning, to end users who deselect the active component (the component selected in the Components pane). To issue a warning, write script code that retrieves the value of this property and displays a message accordingly. Retrieve the value of this property by calling ComponentGetData with the third parameter set to COMPONENT_FIELD_FILENEED.



If you use this property in conjunction with the Required Components property, you may find it complicated to determine which components will be installed as a result of the various possible end-user selections. InstallShield Corporation recommends that you set the value of this property as Critical only for top-level components (that is, components that are not subcomponents of any other component).

Controls

Critical (option button)

End users cannot deselect the component.

Recommended (option button)

End users are advised that the component should be installed.

Standard (option button)

End users may deselect the component; no messages are issued.

{button ,AL(`Required Components page (Component Properties sheet);ComponentGetData I',0,'')} See also

{button ,PI("",`feedback')} Feedback

New command (File menu)

Opens the New dialog box, in which you can create a new project in InstallShield. You can open an existing project by double-clicking its icon in the Projects window.

Shortcuts

Toolbar: 

Keys: Ctrl+N

```
{button ,AL(`General tab (New dialog);Projects tab (New dialog);Projects window;Template tab (New dialog)',0,'')}
```

See also

```
{button ,PI("",`feedback')} Feedback
```

Open command (File menu)

Opens the Open dialog box, in which you can open an existing project by selecting the corresponding setup project file (.ipr file).

Shortcuts

Toolbar: 

Keys: Ctrl+O

{button ,PI("",`feedback')} Feedback

Print Setup command (File menu)

Opens the Print Setup dialog box, where you specify printer properties, paper size and source, and print orientation.

{button ,PI("",`feedback')} Feedback

Print command (File menu)

Opens the Print dialog box, where you may specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options.

Shortcuts

Toolbar: 

Keys: Ctrl+P

{button ,PI("",`feedback')} Feedback

Save command (File menu)

Saves the active project (including all open files at their current state).

Shortcuts

Toolbar: 

Keys: Ctrl+S

{button ,PI("",`feedback')} Feedback

File menu commands

The File menu offers the following commands:

New

Opens the New dialog box, in which you can create a new setup project.

Open

Opens the Open dialog box, in which you can open an existing setup project.

Close

Closes the active setup project.

Save

Saves the active setup project (including all open files at their current state).

Print

Opens the standard Windows Print dialog box for the currently defaulted Windows printer. A header containing the filename and a footer containing the page number and date will be included automatically in your printout.

Print Setup

Opens the standard Windows Print Setup dialog box for the currently defaulted Windows printer. A header containing the filename and a footer containing the page number and date will be included automatically in your printout.

Exit

Closes the active setup project, including all files.

{button ,PI("",`feedback')} Feedback

Format tab (Options dialog box)

Allows you to control screen font (family and size) and the coloring which will be applied to various syntax elements when they appear in your script. All settings made in this tab take effect immediately and are saved as InstallShield defaults for all projects when you exit InstallShield.

Controls

Font (list box)

Select the font in which script text is displayed. All the monospace fonts available on your system (plus Letter Gothic if it is present) are displayed.

Size (list box)

Select the type size in which script text is displayed.

Colors (list box)

Select the syntax element for which you wish to set foreground and background colors.

Foreground (list box)

Displays, and lets you change, the foreground (text) color for the syntax element selected in the Colors list box.

Background (list box)

Displays, and lets you change, the background color for the syntax element selected in the Colors list box.



Selecting Automatic in the Foreground or Background list box causes the corresponding system color to be used.

Reset All (button)

Returns all settings to their default values.

Sample

Displays an example of the font, size, and color settings for the syntax element selected in the Colors list box.

{button ,PI("", `feedback`)} Feedback

Function Name panel (Function Wizard)

Lets you select a built-in InstallShield function to paste into the active script editor window.

Controls

Function Category (list box)

Select a function category; the built-in InstallShield functions belonging to that category are displayed in the Function Name list box. Select "All" to display the names of all built-in InstallShield functions.

Function Name (list box)

Select a function name from among those belonging to the category selected in the Function Category list box.

Next (button)

Go to the Function Parameters panel of the Function Wizard to fill in the function parameters and paste the function call into the active script editor window.

{button ,PI("",`feedback')} Feedback

Function Parameters panel (Function Wizard)

Displays the name of the function selected in the Function Name panel and the parameters for that function. Next to each parameter is either an edit box in which you can enter a variable name or a value, or a drop-down list box from which you can select a parameter value.

{button ,AL(`The Function Wizard I',0,'')} See also

{button ,PI("",`feedback')} Feedback

Function Wizard

Lets you paste a built-in InstallShield function into the active script editor window at the location of the cursor. The Function Wizard consists of the following panels:

Function Name

Lets you select a built-in InstallShield function.

Function Parameters

Lets you set the parameters of the selected function.

{button ,PI("",`feedback')} Feedback

Help menu commands

The Help menu offers the following commands:

[Help Library](#)

Unified access to all InstallShield help.

[Getting Started](#)

Advice and guidelines on beginning to use InstallShield.

[Getting Results](#)

Overviews and step-by-step procedures for all aspects of creating an InstallShield setup.

[Language Reference](#)

A guide to the structure, elements, syntax, and functions of the InstallScript language.

[Quick Preview](#)

A quick preview of InstallShield's integrated development environment.

[Readme](#)

Introductory information such as system requirements, setup instructions, and an overview of InstallShield's documentation.

[About InstallShield](#)

Opens the About InstallShield dialog box.

The InstallShield on the Web submenu offers the following commands:

[Product News](#)

Get up-to-date product information.

[Frequently Asked Questions](#)

Get answers to frequently asked questions about InstallShield.

[Online Support](#)

Request product support or learn about InstallShield support programs.

[Send Feedback](#)

Send feedback such as comments, feature requests, and password requests.

[Register InstallShield](#)

Register your copy of InstallShield.

[InstallShield Home Page](#)

Go to InstallShield's home page.

{button ,PI("",`feedback')} [Feedback](#)

Getting Started command (Help menu)

Opens Getting Started, which provides advice and guidelines on beginning to use InstallShield.

{button ,PI("",`feedback')} Feedback

Getting Results command (Help menu)

Opens Getting Results, which provides overviews and step-by-step procedures for all aspects of creating an InstallShield setup.

{button ,PI("",`feedback')} Feedback

Language Reference command (Help menu)

Opens the InstallScript Language Reference, which explains the structure, elements, syntax, and functions of the InstallScript language.

```
{button ,PI("",`feedback')} Feedback
```

Help Library command (Help menu)

Opens the Help Library, which provides unified access to all InstallShield help. Its Contents, Index, and Find tabs incorporate all InstallShield help files.

{button ,PI("",`feedback')} Feedback

Quick Preview command (Help menu)

Launches a DemoShield demo that provides a quick preview of InstallShield's integrated development environment.

{button ,PI("",`feedback')} Feedback

Media Build Wizard

Lets you build disk images of your software distribution. The Media Build Wizard consists of the following panels:

Name

Lets you name a new build, or select the name of an existing media build to update or rebuild the build.

Disk Type

Lets you select a disk type for your build, and properties of CD-ROM builds.

Build Type

Lets you specify the build type and advanced build properties.

Tag File Information

Lets you store information including your company name, application name, version number, and product category in the tag files placed on your disk images.

Languages

Lets you select the languages your setup supports.

Platforms

{button ,PI("",`OS_limits`)} Visual C++ edition limitation

Lets you select the operating systems your setup supports.

Summary

Displays a summary of build information.

{button ,PI("",`feedback`)} Feedback

Include in Build page (Component Properties sheet)

Lets you specify whether the active component (the component selected in the Components pane) should be included on your distribution media.



You can use this feature to leave 'place holders' for components that aren't yet ready to be included in a build or to exclude components from custom editions of your application.

{button ,PI("",`feedback')} Feedback

Included File Groups page (Component Properties sheet)

Opens when you double-click Included File Groups in the Component Properties window. Lets you view, add to, or remove from the list of file groups associated with the active component (the component selected in the Components pane).

Controls

File Group Name (list box)

Shows all file groups associated with the active component.

Add (button)

Opens the Add File Group dialog box.

Remove (button)

Removes the selected file group from the list in File Group Name. (Disabled if no file group is selected.)

{button ,AL(`Add File Group dialog box')} See also

Insert menu commands

The Insert menu offers the following commands:

File into Script Files

Opens the Insert Files into Script Files dialog box, in which you can select files to be included in the Scripts pane.



When you select a file for insertion, InstallShield makes a copy of the file and places the copy in the <drive name>\My Installations\<project name>\Script Files folder (or the <project name>\Script Files subfolder of the project folder you specified in the Options dialog box's Project Location tab). After that, editing the file in an InstallShield script editor window makes changes to the copy, not the original file. Changes to the original file will have no effect on your setup unless you re-insert the file.

All files displayed in the Scripts pane are located in the above-named folder. Thus, #include statements for those files don't need an explicit path name. For example, if Setup.rul and Functions.rul are displayed in the Scripts pane, include Functions.rul in Setup.rul with the statement `#include "Functions.rul"`.

File into Setup Files

Opens the Insert Files into <setup files folder> dialog box, in which you can select files to be included in the active setup files folder (the folder selected in the Setup Files pane).

Links into File Groups

Opens the Insert File Links to File Group dialog box, in which you can select files to be included in the active Links folder (the Links folder selected in the File Groups pane).

{button ,AL(`Options... command (Tools menu)',0,'')} See also

{button ,PI("",`feedback')} Feedback

File into Script Files command (Insert menu)

Opens the Insert Files into Script Files dialog box, in which you can select files to be inserted into the Scripts pane.



When you select a file for insertion, InstallShield makes a copy of the file and places the copy in the <drive name>\My Installations\
<project name>\Script Files folder (or the <project name>\
Script Files subfolder of the project folder you specified in the
Options dialog box's Project Location tab). After that, editing the file
in an InstallShield script editor window makes changes to the copy,
not the original file. Changes to the original file will have no effect on
your setup unless you re-insert the file.

All files displayed in the Scripts pane are located in the above-
named folder. Thus, #include statements for those files don't need
an explicit path name. For example, if Setup.rul and Functions.rul
are displayed in the Scripts pane, include Functions.rul in Setup.rul
with the statement #include "Functions.rul".

{button ,AL(`Options... command (Tools menu)',0,`,`)} See also

{button ,PI(`",`feedback')} Feedback

File into Setup Files command (Insert menu)

Opens the Insert Files into <setup files folder> dialog box, in which you can select files to be included in the active setup files folder (the folder selected in the Setup Files pane).

{button ,PI("",`feedback')} Feedback

Links into File Groups command (Insert menu)

Opens the Insert File Links to File Group dialog box, in which you can select files to be included in the active Links folder (the Links folder selected in the File Groups pane).

{button ,PI("",`feedback')} Feedback

InstallShield main window

When you launch InstallShield, the Projects window is automatically opened within the main window. When you double-click a project icon in the Projects window, the project workspace for that project opens in the upper left of the main window. The project workspace is open throughout a typical setup creation.

The main window offers menus and, optionally, a toolbar and an output window.

{button ,AL(`Main window menus;Output window;Projects window;Project workspace;toolbar;Getting Started
l',0,`,`')} See also

{button ,PI(`",`feedback')} Feedback

Overwrite page (Component Properties sheet)

Lets you specify the conditions under which the files of the active component (the component selected in the Components pane) overwrite already-existing versions on the target system.

Controls

Overwrite (list box)

Select whether files on the target system are always overwritten, never overwritten, or conditionally overwritten based on date/time stamp or version number.



You may wish to use the "Never overwrite" option for components consisting of files your end user may have updated, such as data and configuration files.

The following controls are enabled only if "Overwrite files BY VERSION" or "Overwrite files BY VERSION THEN DATE (if necessary)" is selected in the Overwrite list box.



If a file on the target system has a version number and the file on your distribution media does not, or vice versa, the file with no version number is treated as if it had a lower version number.

Version: Newer (option button)

A file on the target system is overwritten if the file on your distribution media has a higher version number.

If the file on your distribution media and the file on the target system have the same version number, or if neither file has a version number, and you selected "Overwrite files BY VERSION THEN DATE (if necessary)" from the list box, then whether the file on the target system is overwritten is determined by which Date/Time option you select.

If neither file has a version number and you selected "Overwrite files BY VERSION" from the list box, then the file on the target system is always overwritten.

Version: Newer or Same (option button)

A file on the target system is overwritten if the file on your distribution media has a higher or the same version number.

If neither file has a version number and you selected "Overwrite files BY VERSION THEN DATE (if necessary)" from the list box, then whether the file on the target system is overwritten is determined by which Data/Time option you select.

If neither file has a version number and you selected "Overwrite files BY VERSION" from the list box, then the file on the target system is always overwritten.

Version: Older (option button)

A file on the target system is overwritten if the file on your distribution media has a lower version number.

If the file on your distribution media and the file on the target system have the same version number, or if neither file has a version number, and you selected "Overwrite files BY VERSION THEN DATE (if necessary)" from the list box, then whether the file on the target system is overwritten is determined by which Date/Time option you select.

If neither file has a version number and you selected "Overwrite files BY VERSION" from the list box, then the file on the target system is always overwritten.

The following controls are enabled only if "Overwrite files BY DATE" or "Overwrite files BY VERSION THEN DATE (if necessary)" is selected in the Overwrite list box.

Date/Time: Newer (option button)

A file on the target system is overwritten if the file on your distribution media has a more recent date and time stamp.

Date/Time: Newer or Same (option button)

A file on the target system is overwritten if the file on your distribution media has a more recent or the same date and time stamp.

Date/Time: Older (option button)

A file on the target system is overwritten if the file on your distribution media has a less recent date and time stamp.

{button ,PI("",`feedback')} Feedback

Setup Files pane (project workspace)

The Setup Files pane lets you sort the files that perform your setup according to language, operating system, and location on your distribution medium.

The Setup Files pane contains file folders under the following main headings:

Splash Screen

Place your startup graphic (the graphic displayed at the same time as the startup dialog) in the appropriate language folder.

Language Independent

Place language-independent setup files in the appropriate operating system folder.
{button ,PI("",`OS_limits')} Visual C++ edition limitation

<language>

Place language-dependent setup files in the appropriate operating system folder.

Advanced Files

For multi-disk distributions, place setup files that need to be located on a particular distribution disk in the appropriate disk folder.



When you place a file in the Setup Files pane, InstallShield actually makes a copy of the file and places the copy under the default location: My Installations\<<project name>\Setup Files. Therefore, if you change the source file, those changes will not be automatically included in the copy of the file in your setup project. You must re-insert the file into the Setup Files pane in order for the new file to be included with your project.

{button ,AL(` Access bitmaps during setup I;Display a startup graphic I;Displaying billboards I;Placing resource files in the IDEs Setup Files pane I;Disk ID panel (Media Build Wizard)',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Project workspace

The project workspace contains the following panes, each of which controls a different portion of the setup creation process:

[File Groups](#)

Controls file groups.

[Components](#)

Controls components.

[Setup Types](#)

Controls the setup types you offer your end user.

[Scripts](#)

Controls script files.

[Resources](#)

Controls string resources used in your setup.

[Setup Files](#)

Controls the files that perform the setup on your end user's system.

[Media](#)

Controls the building of disk images.

{button ,PI("",`feedback`)} [Feedback](#)

Keys tab (Options dialog box)

Lets you assign or remove keyboard commands for editing functions.

Controls

Commands (list box)

Specifies the command to which the keyboard shortcut(s) in the "Current keys" list box are assigned.

Assign (button)

Assigns the keyboard shortcut in the "Press new shortcut" edit box to the command selected in the Commands list box. Disabled if there is no keyboard shortcut in the "Press new shortcut" edit box, or if the keyboard shortcut in the "Press new shortcut" edit box is already assigned to another command. In the latter case, you are shown (right below the "Press new shortcut" edit box) what command the keyboard shortcut is assigned to.

Remove (button)

Removes the keyboard shortcut selected in the "Current keys" list box.

Reset All (button)

Resets keyboard shortcuts for all commands to their default values.

Press new shortcut (edit box)

Enter the keyboard shortcut to be assigned to the command selected in the Commands list box; press the keys, rather than typing their names. (Assign the shortcut by clicking the Assign button.)

Current keys (list box)

Displays the keyboard shortcut(s) currently assigned to the command selected in the Commands list box.

{button ,PI("", `feedback`)} [Feedback](#)

Languages panel (Media Build Wizard)

Lets you select the languages your setup will support. Files specific to the selected languages and language-independent files are included in the file media library; files specific to deselected languages are not included in the file media library. To select multiple languages, hold down the Shift or Ctrl key and click with the mouse.

Double-click a language to make it the default language for the setup.



Clicking the Cancel button results in no build being created, and no existing build being overwritten.

{button ,PI("",`feedback`)} Feedback

InstallShield for Visual C++ does not include this function.

Language page (File Group Properties sheet)

Lets you specify to what language(s), if any, the active file group (the file group selected in the File Groups tab) is specific.

You can use this property to determine which file groups are included in your file media library, and which are installed on your end user's hard drive.

- In the Media Build Wizard's Languages panel, you can specify particular languages to which the build is specific; file groups specific to other languages will not be included in the build.
- In your script, you can allow or prevent installation of file groups that are specific to particular languages using the ComponentFilterLanguage function.

Controls

Languages (list box)

Select the language(s), if any, to which the active file group is specific.

Show sublanguages (check box)

If checked, shows variants of base languages, for example, Norwegian (Bokmal) and Norwegian (Nynorsk).

Show only installed languages (check box)

If checked, the only languages shown are those for which you have purchased InstallShield resources.

{button ,AL(`Languages panel (Media Build Wizard);ComponentFilterLanguage I',0,';')} See also

{button ,PI("",`feedback')} Feedback

Maximize command (Control menu)

Enlarges the active window to fill the available space.

Shortcut

Mouse: Click the maximize icon  on the title bar; or double-click the title bar.

{button ,PI("",`feedback')} Feedback

Media pane popup menu

Build Media Wizard

Launch the Media Build Wizard.

Send Media To

Launch the Send Media Wizard.

Delete

Delete the selected build.

Rename

Rename the selected build.

{button ,AL(`Media Build Wizard;Send Media Wizard',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Media pane (project workspace)

The Media pane displays a Media Build Wizard icon, and icons for already built file media libraries. Double-clicking the Media Build Wizard icon launches the Media Build Wizard, which displays a series of panels that help you construct your file media library.

Below the icons for already existing builds are folders for the disk images and for report files and log files from all builds under the same build name. Report files are summaries of selected build options; log files contain the messages generated by the file media library builder.



InstallShield cannot handle project file paths with more than 260 characters (including the file name). The names of the following project elements are incorporated in project paths and file names:

- n the project itself
- n file groups
- n media names
- n files in the Scripts pane
- n files inserted into the Setup Files pane

To help you keep within the 260-character limit, InstallShield warns you if you attempt to assign a name with more than 50 characters to a project, file group, or media name.

{button ,PI("", `bldrpt')} [Visual C++ edition limitation](#)

{button ,AL(`Media pane popup menu',0,`,`')} [See also](#)

{button ,PI("", `feedback')} [Feedback](#)

InstallShield for Visual C++ does not generate build reports.

Main window menus

The main InstallShield window offers the following menus:

File

Open, close, and save projects; print script files.

Edit

Edit script files.

View

Enable and disable display of various windows.

Insert

Insert files or file links in various folders.

Build

Compile, run, debug, and build setups.

Tools

Set options for script editor windows; launch the Windows Explorer.

Window

Control display of windows.

Help

Get help for InstallShield.

{button ,PI("",`feedback')} Feedback

Tools menu commands

The Tools menu offers the following commands:

[Options](#)

Opens the Options dialog box.

[Launch Explorer](#)

Launches the Windows Explorer.

[Dialog Sampler](#)

Launches the Dialog Sampler.

{button ,PI(,"",`feedback')} Feedback

Dialog Sampler command (Tools menu)

Launches the Dialog Sampler, in which you can view any InstallShield dialog box.



When you launch InstallShield International's Dialog Sampler, the [Choose Setup Language dialog box](#) is displayed, offering you a choice of languages in which to display the InstallShield dialog boxes. For the dialog boxes to be displayed correctly, the language you select must use the same Windows code page as the language in which your system is running.

{button ,PI("",`feedback')} Feedback

Move command (Control menu)

Displays a four-headed arrow  so you can move the active window or dialog box with the arrow keys. This command is unavailable if you maximize the window.

{button ,PI("",`feedback')} Feedback

Media Name panel (Media Build Wizard)

Lets you name a new build, or select the name of an existing build to update or rebuild the build.



InstallShield cannot handle project file paths with more than 260 characters (including the file name). The names of the following project elements are incorporated in project paths and file names:

- n the project itself
- n file groups
- n media names
- n files in the Scripts pane
- n files inserted into the Setup Files pane

To help you keep within the 260-character limit, InstallShield warns you if you attempt to assign a name with more than 50 characters to a project, file group, or media name.



Clicking the Cancel button results in no build being created, and no existing build being overwritten.

New dialog box

Opens when you select the New command from the File menu. Lets you select a project or project template to copy as a new project. Contains the following tabs:

[General](#)

[Projects](#)

[Template](#)

{button ,PI("",`feedback')} Feedback

New Window command (Window menu)

Opens a new window with the same contents as the active window. You can open multiple windows to display different parts or views of the window contents at the same time. If you change the contents in one window, all duplicate windows reflect those changes. When you open a new window, it becomes the active window and is displayed on top of all other open windows.

{button ,PI("",`feedback')} Feedback

Next command (Control menu)

Switches to the next open window. InstallShield determines which window is next according to the order in which you opened the windows.

Shortcut

Keys: Ctrl+F6

{button ,PI("",`feedback')} Feedback

No Help Available

No help is available for this area of the window.

{button ,PI("",`feedback')} Feedback

No Help Available

No help is available for this message box.

{button ,PI("", `feedback')} Feedback

Operating System page (File Group Properties sheet)

Lets you specify to what operating system(s) if any, the active file group (the file group selected in the File Groups tab) is specific. To select multiple operating systems, hold down the Shift or Ctrl key and click with the mouse.

{button ,PI("",`OS_limits')} Visual C++ edition limitation.

You can use this property to determine which file groups are included in your file media library.

- In the Media Build Wizard's Platforms panel, you can specify particular operating systems to which the build is specific; file groups specific to other operating systems will not be included in the build.
- In your script, you can allow or prevent installation of file groups that are specific to particular operating systems using the ComponentFilterOS function.



16-bit platforms such as Windows 3.1 can't read long filenames. If the platforms targeted by your setup include a 16-bit platform, don't use long filenames for your application files.

{button ,AL(`Platforms panel (Media Build Wizard);ComponentFilterOS l',0,','')} See also

{button ,PI("",`feedback')} Feedback

Options dialog box

Opens when you select the Options command from the tools menu. Lets you set script editor window properties, text formatting, and tab stops. Contains the following tabs:

[Edit](#)

[Format](#)

[Keys](#)

[Project Location](#)

[Tabs](#)

{button ,PI("",`feedback')} [Feedback](#)

Options command (Tools menu)

Opens the Options dialog box, letting you set script editor window properties, text formatting, and tab stops.

{button ,AL(`Edit tab (Options dialog);Format tab (Options dialog);Tabs tab (Options dialog);Keys tab (Options dialog);Project Location tab (Options dialog)',0,`,`')}} See also

{button ,PI(`",`feedback')}} Feedback

Output window

The Output window, at the bottom of the InstallShield main window, displays messages generated during script compilation and file media library building.

{button ,AL(`Output window popup menu',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Password page (Component Properties sheet)

Lets you specify a password for the active component (the component selected in the Components pane). You can specify a different password for each different component. You can also specify a password for the entire setup in the Password page of the Media Build Wizard's Advanced Build Type Properties sheet.

To compare this password with your end user's input during the setup, call the ComponentValidate function in your script.



Experienced hackers may be able to extract this password from your distribution media. If security is a critical issue, InstallShield Corporation recommends that you not depend solely on this password protection feature.

Controls

Current Password (static text field)

Displays the current password.

New Password (edit box)

Enter a new password for the active component (the component selected in the Components pane).



Password protection is not the same as encryption. Password protecting a component prevents it from being installed by someone who does not know the password. Encrypting a file scrambles it so that if someone opens it in a text editor, they cannot read strings and other data to get confidential information about your application. You can password protect a component or you can encrypt the component's files—or, for maximum security, you can do both.

If a component's files are encrypted but neither the component nor the setup is password protected, the component can be installed by anyone who obtains your distribution media (or a copy). If a component is password protected but its files are not encrypted, someone can open the files in a text editor and read confidential information—possibly including the component's password.

{button ,AL(`ComponentValidate |Encryption page (Component Properties sheet);Password page (Advanced Build Type property sheet);Get and validate a component or media librarys password |',0,'')} See also

{button ,PI("",`feedback')} Feedback

Password page (Advanced Build Type property sheet)

Lets you specify a password for the entire setup. You can specify separate passwords for individual components in the Password pages of the components' property sheets.

To compare this password with your end user's input during the setup, call the ComponentValidate function in your script.



Experienced hackers may be able to extract this password from your distribution media. If security is a critical issue, InstallShield Corporation recommends that you not depend solely on this password protection feature.

Controls

Use password (check box)

If checked, the text in the Password edit box is the password for the build; if unchecked, the build is not password protected.

Password (edit box)

Enter a password for the build. (Disabled if the "Use password" check box is unchecked.)

```
{button ,AL(`Password page (Component Properties sheet);ComponentValidate I',0,';')} See also  
{button ,PI("",`feedback')} Feedback
```

Paste command (Edit menu)

Use this command to insert a copy of the clipboard contents at the insertion point. This command is unavailable if the clipboard is empty.

Shortcuts

Toolbar: 

Keys: Ctrl+V

{button ,PI("",`feedback')} Feedback

Platforms panel (Media Build Wizard)

Lets you select the operating systems your setup will support. Files specific to the selected operating systems and OS-independent files are included in the build; files specific to deselected operating systems are not included in the build. To select multiple operating systems, hold down the Shift or Ctrl key and click with the mouse.

{button ,PI("",`OS_limits')} Visual C++ edition limitation.



16-bit platforms such as Windows 3.1 can't read long filenames. If the platforms targeted by your setup include a 16-bit platform, don't use long filenames for your application files.



Clicking the Cancel button results in no build being created, and no existing build being overwritten.

{button ,PI("",`feedback')} Feedback

Potentially Locked page (File Group Properties sheet)

Lets you specify whether already-installed versions of the files in the active file group (the file group selected in the File Groups tab) may be locked—that is, in use by another application—during setup.

{button ,PI("",`feedback')} Feedback

Print Progress dialog box

The Printing dialog box is shown during the time that InstallShield is sending output to the printer. The page number indicates the progress of the printing.

To abort printing, choose Cancel.

{button ,PI("", `feedback')} Feedback

Product Files dialog box

Opens when you click the Files button in the About InstallShield dialog box. Displays version numbers for the files provided with your copy of InstallShield.

Controls

Name/Version (list box)

Lists names and version numbers of all files in the category folder selected in the tree control. The files in the Program Files folder are the files that make up InstallShield itself; the files in the Redistributable Files folder's subfolders are the files you distribute with your setup.

Copy All (button)

Copies name and version information, for all files in all file categories, to the clipboard in plain text format.

{button ,AL(^About InstallShield dialog box',0,'')} See also

{button ,PI("",`feedback')} Feedback

Projects window

The Projects window contains the Project Wizard icon, and icons for already existing setup projects. Double-clicking one of the project icons opens the project workspace for that project.

{button ,AL('Project Wizard;Project workspace;Projects window popup menu;Getting Started I',0,'')} See also

{button ,PI('','feedback')} Feedback

Projects window popup menu

Open

Opens the selected project.

Project Wizard

Launches the Project Wizard.

Delete

Deletes the selected project. Does not delete your application files.

Rename

Changes the project name displayed in the Projects window.

Move

Opens the Select Target Folder for Project dialog box, in which you can specify a folder to which all of the selected project's files are moved.

Run

Compiles and runs the selected project.

Debug

Compiles and runs the selected project in the InstallShield Visual Debugger.

Properties

Displays the selected project's Project Properties sheet.

{button ,AL(` debug |;Setup Properties sheet;Project Wizard',0,'')} See also

{button ,PI("",`feedback')} Feedback

Replace dialog box

Opens when you select Replace from the Edit menu. Lets you replace text in the active script file (the one selected in the Scripts pane).

Controls

Find what (combo box)

Enter the text to search for, or select search text from the drop-down list box.

Replace with (combo box)

Enter the text to replace the "Find what" text, or select replacement text from the drop-down list box.

Match whole word only (check box)

If checked, finds only whole words; if unchecked, also searches for your text inside longer words.

Match case (check box)

If checked, only text with the same pattern of upper and lower case letters that you specified in "Find what" is found.

Replace in: Selection (option button)

Perform the text replacement only within the selected text. (If there is no selected text, this button is disabled.)

Replace in: Whole file (option button)

Perform the text replacement within the entire file.

Find Next (button)

Finds the next occurrence of the text in "Find what."

Replace (button)

Replaces the current instance of found text with the text in "Replace with." (If no text has been found, this button is disabled.)

Replace All (button)

Replaces all occurrences of the text in "Find what" with the text in "Replace with."

{button ,PI("", `feedback`)} [Feedback](#)

Required Components page (Component Properties sheet)

Lets you add to or remove from the list of components that must be installed if the active component (the component selected in the Components pane) is installed.



If you use this property in conjunction with the File Need property, you may find it complicated to determine which components will be installed as a result of the various possible end-user selections.

Controls

Required Components (list box)

Lists the components currently designated as required by the active component.

Components (list box)

Lists all defined components. The active component has a check mark.

Add (button)

Adds the component selected in the Components list box to the Required Components list box.

Remove (button)

Removes the selected component from the Required Components list box.

{button ,AL('File Need page (Component Properties sheet)',0,'')} See also

{button ,PI('','feedback')} Feedback

Resources pane (project workspace)

Displays a tree control for setup resources: string resources, registry entries, and shell objects (folders/groups and shortcuts/icons).

String resources

Clicking a string table's icon opens the String Table window, in which you can view and modify the contents of that string table.

Registry entries

Right-clicking the Registry Entries folder opens a popup menu from which you can create a new registry set.

Clicking a registry set icon opens the Registry Entries window, in which you can view and modify the keys and values in that registry set.

Shell objects

Right-clicking the icon of a predefined shell object (folder, group, or Program Manager Shell) opens a popup menu from which you can create a new folder or shortcut (under the Explorer Shell) or group or icon (under the Program Manager Shell). Clicking the icon of a folder or group (or shortcut or icon) opens the Folders window (or the Shortcuts window), in which you can view and modify the properties of that folder or group (or shortcut or icon).



The name of a setup resource must not contain any characters that are invalid in folder names. In Windows 95, invalid characters are the following:

\\ : * ? " < > |

{button ,AL('Registry Entries window;Shell Objects - Folders window;Shell Objects - Shortcuts window;Strings window',0,'')} See also

{button ,PI("",`feedback`)} Feedback

Restore command (Control menu)

Returns the active window to its size and position before you chose the Maximize or Minimize command.

{button ,PI("",`feedback')} Feedback

Self Registered page (File Group Properties sheet)

Lets you specify whether the files in the active file group (the file group selected in the File Groups tab) are self-registering.



If any files in the file group are not self-registering, but you select the "Yes" option button in this property page, an error results when the setup runs.



unInstallShield will unregister the files only if the file group is marked as both self-registered and shared.

Controls

Yes (option button)

Select this option if the files are self-registering and you want InstallShield to register them during the setup.

No (option button)

Select this option if the files are not self-registering or you do not want InstallShield to register them during the setup.

{button ,AL('Install self-registering files I;Shared page (File Group Properties sheet)',0,'')} See also
{button ,PI('','feedback')} Feedback

Send Media Wizard

The Send Media Wizard lets you send your disk images to another folder on your system or to disks in a disk drive. The Send Media Wizard consists of the following panels:

Select Media

Select the file media library to be sent.

Select Type

Select whether to send to another folder or to a disk drive.

Depending on your selection in the Select Type panel, you will see one of the following two panels:

Select Removable Drive

Select the drive letter.

Target Folder

Select the target folder.

Status

Monitor the status of the file media library transfer.

{button ,PI("", `feedback`)} Feedback

Select Media panel (Send Media Wizard)

From the list box, select the name of the file media library you wish to send to another folder or to a disk drive.

{button ,PI("",`feedback')} Feedback

Select Type panel (Send Media Wizard)

Select whether to send the disk images to another folder on your system or to disks in a disk drive.

{button ,PI("",`feedback')} Feedback

Select Removable Drive panel (Send Media Wizard)

From the drop-down list box, select the letter of the disk drive to which you want to send the disk images. (This panel displays when you select the "Copy each 'disk' in the media to a removable disk" option button in the Select Type panel and click the Next> button.)

{button ,PI("",`feedback')} Feedback

Target Folder panel (Send Media Wizard)

Specify the folder to which you want to copy the disk images. (This panel displays when you select the "Copy the media files to a fixed folder" option button in the Select Type panel and click the Next> button.)

Controls

Target Folder (edit box)

Enter the path of the folder to which you want to copy the disk images.

Browse (button)

Opens the Select Folder dialog box, in which you can select the folder to which you want to copy the disk images.

{button ,PI("",`feedback')} Feedback

Status panel (Send Media Wizard)

Displays the status of the file media library transfer.

Controls

Start (button)

Starts the file media library transfer.

{button ,PI("",`feedback')} Feedback

Setup page (Advanced Build Type property sheet)

Lets you modify the startup message dialog box or display the Choose Setup Language dialog box (dialog boxes shown below).



Controls

Application Name (edit box)

Enter the application name to display in the startup message: "<application name> Setup is preparing the InstallShield Wizard which will guide you through the rest of the setup process. Please wait."

Enable Language Dialog (check box)

If checked, displays the Choose Setup Language dialog box before displaying the startup message dialog box. The list box in this dialog box displays all the languages that you select in the Media Build Wizard's Languages panel. The language you specify as the default in the Languages panel is the language initially displayed in the dialog box's read-only text box.

{button ,AL(`Languages panel (Media Build Wizard)',0,'')} See also

{button ,PI("",`feedback')} Feedback

Project Properties sheet

Opens when you select Properties from the Projects window's popup menu or click the Properties button in the New dialog box. Displays information about the selected setup project; if opened from the Projects window, lets you modify the project. Contains the following pages:

[General](#)

[Info](#)

[Instructions](#)

[Language](#)

[Operating Systems](#)

[Owner](#)

{button ,PI("", `feedback')} [Feedback](#)

Setup Types pane popup menu

New Setup Type

Create a new setup type.

Delete

Delete the selected setup type.

Rename

Rename the selected setup type.

Properties

Opens the Setup Type dialog box.

`{button ,AL(`Setup Type dialog box',0,'')}_` See also

`{button ,PI("",`feedback')}_` Feedback

Setup Types pane (project workspace)

Displays existing setup types. Clicking a setup type opens the Setup Types window, displaying components associated with that setup type.



The SetupType function displays only the standard setup types Typical, Compact, and Custom, with fixed description text for each. If you want greater flexibility, call SdSetupTypeEx in your script instead.

(The Project Wizard puts an SetupType call in the script if you accept the default selections in the Specify Setup Types panel.)

{button ,AL(` Setup Type window;Setup Type pane popup menu;SdSetupTypeEx l;SetupType l',0,'','')} See
also

{button ,PI('','feedback')} Feedback

Project Wizard

The Project Wizard lets you quickly create a new setup project, including a compilable script. The Project Wizard consists of the following panels:

Welcome

Enter information about your application.

Choose Dialogs

Select the built-in InstallShield dialog boxes you want to use in your setup.

Choose Target Platforms

{button ,PI("",`OS_limits')} Visual C++ edition limitation

Select the operating systems on which your setup runs.

Specify Languages

Select the languages in which your setup is available.

Specify Setup Types

Select the setup type options to offer your end users.

Specify Components

{button ,PI("",`comp_limits')} Visual C++ edition limitation

Select the components into which your application is organized.

Specify File Groups

{button ,PI("",`fg_limits')} Visual C++ edition limitation

Select the file groups into which your application's files are organized.

Summary

Review your selections.

{button ,PI("",`feedback')} Feedback

Shared page (File Group Properties sheet)

Lets you specify whether the files in the active file group (the file group selected in the File Groups tab) are shared files—that is, files (for example, drivers and system DLLs) that may be used by more than one application. If you specify the file group's files as shared, InstallShield automatically handles the files' registry reference counters during the setup.

{button ,PI("",`feedback')} Feedback

Size command (Control menu)

Displays a four-headed arrow so you can size the active window with the arrow keys. This command is unavailable if you maximize the window.



After the pointer changes to the four-headed arrow:

1. Press one of the direction keys (left, right, up, or down arrow key) to move the pointer to the border you want to move.
2. Press a direction key to move the border.
3. Press Enter when the window is the size you want.

Shortcut

Mouse: Drag the size bars at the corners or edges of the window.

{button ,PI("", `feedback')} Feedback

Scripts pane popup menu

New File

Creates a new script file.

Paste

Pastes a script file copied from the Windows Explorer.

Delete

Deletes the selected script file.

Rename

Renames the selected script file.

Insert Files

Opens the Insert Files into Script Files dialog box.



When you select a file for insertion, InstallShield makes a copy of the file and places the copy in the <drive name>\My Installations\
<project name>\Script Files folder (or the <project name>\
Script Files subfolder of the project folder you specified in the
Options dialog box's Project Location tab). After that, editing the file
in an InstallShield script editor window makes changes to the copy,
not the original file. Changes to the original file will have no effect on
your setup unless you re-insert the file.

All files displayed in the Scripts pane are located in the above-
named folder. Thus, #include statements for those files don't need
an explicit path name. For example, if Setup.rul and Functions.rul
are displayed in the Scripts pane, include Functions.rul in Setup.rul
with the statement `#include "Functions.rul"`.

Properties

Opens the Script Properties sheet for the selected script file.

{button ,AL(`Options... command (Tools menu)',0,`,`')} See also

{button ,PI(`",`feedback')} Feedback

Scripts pane (project workspace)

Displays icons for each script file associated with the active project. To open a script file, double-click its icon; a script editor window opens in which you can view or modify the file.



InstallShield cannot handle project file paths with more than 260 characters (including the file name). The names of the following project elements are incorporated in project paths and file names:

- n the project itself
- n file groups
- n [media names](#)
- n files in the Scripts pane
- n files inserted into the Setup Files pane

To help you keep within the 260-character limit, InstallShield warns you if you attempt to assign a name with more than 50 characters to a project, file group, or media name.

{button ,AL(`Script editor window;Scripts pane popup menu',0,'')} See also

{button ,PI("",`feedback')} Feedback

Specify Components panel (Project Wizard)

Displays a list of the [components](#) included in your setup, and lets you add to or delete from that list.

{button ,PI('interfac.hlp','comp_limits')} Visual C++ edition limitation

Controls

Add (button)

Adds a new component to the list of included components. When the component is added, its name is editable.

Delete (button)

Deletes the selected component from the list of included components.

{button ,PI("",`feedback')} Feedback

Specify File Groups panel (Project Wizard)

Displays a list of the [file groups](#) included in your setup, and lets you add to or delete from that list.

{button ,PI('interfac.hlp','fg_limits')} Visual C++ edition limitation

Controls

Add (button)

Adds a new file group to the list of included file groups. When the file group is added, its name is editable.

Delete (button)

Deletes the selected file group from the list of included file groups.

{button ,PI("",`feedback')} Feedback

Specify Languages panel (Project Wizard)

Lets you select the languages you want your setup to offer.

Controls

Languages (list box)

Select the languages you want your setup to offer. To select multiple languages, hold down the Shift or Ctrl key and click with the mouse.

Show only available languages (check box)

If checked, the only languages shown are those for which you have purchased InstallShield resources.

Show Sublanguages (check box)

If checked, shows variants of base languages, such as Norwegian (Bokmal) and Norwegian (Nynorsk).

{button ,PI("", `feedback`)} [Feedback](#)

Specify Setup Types panel (Project Wizard)

Lets you select the setup types included in your setup.



If you select only the standard setup types Typical, Compact, and Custom (all three are selected by default), the Project Wizard puts a call to the SetupType function in the script. If you select any other setup types, the Project Wizard puts a call to the SdSetupTypeEx function in the script.

{button ,AL(`Setup Type pane (project workspace);SdSetupTypeEx I;SetupType I',0,`,`;`)} See also

{button ,PI(``,`feedback`)} Feedback

Status Text page (Component Properties sheet)

Lets you add, view, or modify the status text for the active component (the component selected in the Components pane). Your end users will see this text in the uppermost line of the progress indicator during the transfer of this component.



To display a progress indicator during your setup, you must enable the progress indicator with the Enable function.

Controls

Current Text (static text field)

Displays the current status text.

New Text (edit box)

When you open the Status Text page, the New Text edit box is filled with the current status text (which is also shown in the Current Text box). If you enter new text in the New Text edit box, the status text is replaced.



If the status text is set to a null string, during the setup the progress indicator will display the status text from the previously installed component. So you can specify status text for a component, and specify no status text for any of its subcomponents; then the same status text will display during the installation of each subcomponent.

>> (button)

Opens the String Entry dialog box. In the String Entry dialog box, click the OK button to specify the string resource in the String Identifier list box as the component description text.

{button ,AL(`String Entry dialog box;Enable I',0,'')} See also

{button ,PI("",`feedback')} Feedback

String Entry dialog box

Lets you view, add, or modify string resources in your project.

Controls

String Identifier (list box)

Choose the identifier of an existing string resource.

Value (static text field)

Displays the text of the string.

Comment (static text field)

Displays comments regarding the string.

Add (button)

Opens the Add/Modify String Entry dialog box.

Modify (button)

Opens the Add/Modify String Entry dialog box.

{button ,AL(`Add/Modify String Entry dialog box',0,`,`)} See also

{button ,PI(`",`feedback')} Feedback

Add/Modify String Entry dialog box

Lets you add or modify string resources in your project.

Controls

String Identifier (edit box)

Displays the identifier of the string resource you're modifying, or a default name (which you can change) for the string resource you're adding.

Value (edit box)

Lets you add or modify the text of the string.

Comment (edit box)

Lets you add or modify comments regarding the string.



If a string is used in the script to name a folder, the string must not contain any characters that are invalid in folder names. For example, the Project Wizard-generated script uses the COMPANY_NAME string to name a folder.

In Windows 95, invalid characters are the following:
\\ : * ? " < > |

{button ,PI("",`feedback')} Feedback

Summary panel (Media Build Wizard)

Displays a summary of the selections you've made in the previous panels of the Media Build Wizard. Clicking the Finish button begins the build. To change a build setting, click the Back button until you return to the relevant panel.



Clicking the Cancel button results in no build being created, and no existing build being overwritten.

{button ,PI("",`feedback')} [Feedback](#)

Summary panel (Project Wizard)

Displays a summary of the selections you've made in the previous panels of the Project Wizard. Clicking the Finish button begins the creation of the project. To change a project setting, click the Back button until you return to the relevant panel.

{button ,PI("",`feedback')} Feedback

Tabs tab (Options dialog box)

Lets you control tabs.

Controls

File Type (list box)

Lets you select Script File as the kind of file to which the settings will apply. Other types of files may be added in future versions of InstallShield.

Tab size (edit box)

Set the tab size as a multiple of the character space size.

Insert spaces (option button)

Tabs are converted to the number of spaces specified in the "Tab size" edit box.

Keep tabs (option button)

Tab characters entered with the Tab key remain tab characters (ASCII 08).

Auto indent: None (option button)

Pressing Enter places the cursor at the left margin in a new line.

Auto indent: Default (option button)

Pressing Enter places the cursor at the same tab stop where the previous line's text begins.

{button ,PI("",`feedback')} Feedback

Minimize command (Control menu)

Reduces the active window to an icon at the bottom of the InstallShield main window, or the main window to an icon on the Windows taskbar.

Shortcut

Mouse: Click the minimize icon  on the title bar.

{button ,PI("",`feedback')} Feedback

Tag File panel (Media Build Wizard)

Lets you store information including your company name, application name, version number, product category, and additional text in the [tag files](#) that InstallShield automatically places on your disk images.



Clicking the Cancel button results in no build being created, and no existing build being overwritten.

{button ,PI("",`feedback')} Feedback

Files used to identify particular distribution disks or disk image folders.

Destination page (Component Properties sheet)

Lets you specify the destination folder to which the active component (the component selected in the Components pane) is transferred.

Controls

Destination (tree control)

Select the destination folder to which the component is transferred.

Current Value (static text field)

Displays the current value of the folder selected in the tree control.

New Folder (button)

Create a new folder below the folder selected in the tree control.

Rename (button)

Rename the folder selected in the tree control. (Enabled only for folders you have added with the New Folder button.)

Delete (button)

Delete the folder selected in the tree control. (Enabled only for folders you have added with the New Folder button.)



The value of an entry under "Script Defined Folders" must be the script variable surrounded by angle brackets, for example, <szFolder>. Access this value in the script by using the ComponentSetTarget function.



The value of "General Application Destination," TARGETDIR, is set in the script with an assignment statement.



The target folders of subcomponents can be different from the target folder of their parent component. In this case, files associated with a subcomponent will be transferred to the subcomponent's target folder, not the parent component's target folder. Only those files associated directly with the parent component, not any of its subcomponents, will be transferred to the parent component's target folder.

{button ,AL('ComponentSetTarget ' ,0,'')} See also

{button ,PI('','feedback')} Feedback

CD-ROM Folder page (Component Properties sheet)

Lets you specify the folder on the distribution media in which the active component (the component selected in the Components pane) is placed if the "Data as files" check box is checked in the Media Build Wizard's Disk Type panel. If no folder is specified and the "Data as files" check box is checked, the component is placed on the root of the distribution media.

This is typically a relevant option only for CD-ROM distributions; for other distribution media, you usually want to compress all components into cabinet files.



Folder names on CD-ROMs should conform to ISO 9660 (Level 1) standards. That is, folder names should be no more than eight characters long (with no extensions), and use only the characters A-Z, 0-9, and _ (underscore).



By putting a folder structure on your CD-ROM, you can allow your application to be run from the CD-ROM. The folder structure on the CD-ROM need not mirror the folder structure that would be installed to a hard drive. All that is necessary is that your application be able to work with either folder structure.

{button ,AL(`Disk Type panel (Media Build Wizard)',0,`,`)} See also

{button ,PI(`",`feedback')} Feedback

Tile command (Window menu)

Arranges multiple opened windows in a non-overlapped fashion.

{button ,PI("",`feedback')} Feedback

Undo command (Edit menu)

Reverses the last editing action, if possible. The Undo command is disabled if you cannot reverse your last action.

Shortcuts

Keys: Ctrl+Z or Alt+Backspace

{button ,PI("",`feedback')} Feedback

Status Bar command (View menu)

Toggles the display of the status bar. When checked, the status bar is shown; when unchecked, the status bar is hidden.

{button ,AL(`status bar',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Toolbar command (View menu)

Toggles the display of the toolbar. When checked, the toolbar is shown; when unchecked, the toolbar is hidden.

{button ,AL(`toolbar',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

View menu commands

The View menu lets you show or hide various parts of the InstallShield window. Checked items are shown; unchecked items are hidden. The View menu offers the following commands:

Toolbar

Shows or hides the toolbar.

Status Bar

Shows or hides the status bar.

Project Workspace

Shows or hides the project workspace.

Output

Shows or hides the output window.

Projects

Shows or hides the Projects window.

{button ,PI("", `feedback`)} Feedback

Project Workspace command (View menu)

Toggles the display of the project workspace. When checked, the project workspace is shown; when unchecked, the project workspace is hidden.

{button ,AL('Project workspace',0,'')} See also

{button ,PI("",`feedback')} Feedback

Output command (View menu)

Toggles the display of the output window. When checked, the output window is shown; when unchecked, the output window is hidden.

{button ,AL(`Output window',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Projects command (View menu)

Toggles the display of the Projects window. When checked, the Projects window is shown; when unchecked, the Projects window is hidden.

{button ,AL('Projects window',0,'')} See also

{button ,PI("",`feedback')} Feedback

Visible page (Component Properties sheet)

Lets you specify whether the active component (the component selected in the Components pane) will be visible in component dialog boxes during the setup. This property does not affect the actual installation of the component—just its display in component dialog boxes during the setup.



Since the only way for your end user to change the selection status of an invisible component is by selecting or deselecting a visible component that requires the invisible component, you should make every invisible component a required component for one or more visible components. Otherwise, initially deselected invisible components can't be selected by the end user and are never installed; and initially selected invisible components let your end users deselect all components and still see non-zero required space.

{button ,AL('Mark a component as critical highly recommended or standard I;Required Components page (Component Properties sheet)',0,'')} See also

{button ,PI('','feedback')} Feedback

Welcome panel (Project Wizard)

Lets you enter basic information about your application.

Application Name (edit box)

Enter the name of your application. InstallShield uses this name in the following places:

- n Registry entries for your application.
- n The title and caption bar of your main setup window.
- n Certain dialog boxes in your setup.

Company Name (edit box)

Enter the name of your company. InstallShield uses this name in the registry entries for your application.

Development Environment (list box)

Select the environment in which you developed your application.

Application Type (list box)

Select the list item that best categorizes your application.

Application Version (edit box)

Enter the version number of your application.

Application Executable (edit box)

Enter the name of the main executable file for your application, or click the ... button to search for the file.

... (button)

Opens the Open dialog box, in which you can select the main executable file for your application.

{button ,PI("", `feedback')} Feedback

Arrange Icons command (Window menu)

Arranges the icons for minimized windows at the bottom of the main window.



If there is an unminimized window at the bottom of the main window, then some or all of the icons may not be visible because they will be underneath this unminimized window.

{button ,PI("",`feedback')} Feedback

Window menu commands

The Window menu offers the following commands, which enable you to arrange windows within the InstallShield window:

New Window

Creates a new window that views the same contents as the active window.

Cascade

Arranges windows in an overlapped fashion.

Tile

Arranges windows in non-overlapped tiles.

Arrange Icons

Arranges icons of minimized windows.

Close All

Closes all windows except the Projects window, the project workspace, and the output window.

Window 1, 2, ...

Makes specified window the active window. The active window is checked.

{button ,PI("",`feedback`)} Feedback

A self-registering file is an OLE server that can place information about itself in the registry so that it is seen by other OLE applications. (Such a file can also remove this information from the registry when it is uninstalled.)

Status bar



The status bar is displayed at the bottom of the InstallShield window. To display or hide the status bar, use the Status Bar command in the View menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as the pointer passes over the buttons.

The status bar displays information about the state of your computer and the program.

- OVR or INS display whether the current typing mode is Overtyping or Insert. (The mode can be changed by pressing the Insert key.)
- CAP displays if the Caps Lock key is latched down.
- NUM displays if the Num Lock key is latched down.

Editing references

When you are editing a script, the status bar also displays the line number and left-right "column position" of your cursor.

Line numbers are counted from the first carriage return in your script. Column positions represent the number of characters, including spaces and the spaces implied by tab and indent settings, from the first available typing position at the left edge of your screen.

{button ,PI("",`feedback')} [Feedback](#)



Toolbar



The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in InstallShield. Most of these tools are also available in the InstallShield menus.

To hide or display the toolbar, choose **Toolbar** from the **View** menu (Alt+V,T).

Toolbar buttons



Opens a new project. (Same as [File|New.](#))



Opens an existing project. (Same as [File|Open.](#))



Saves the active project with its current name. (Same as [File|Save.](#))



[Edit|Cut.](#)

Removes selected data from the project and stores it on the clipboard. (Same as



[Edit|Copy.](#))



Inserts the contents of the clipboard at the insertion point. (Same as [Edit|Paste.](#))



Launches the Media Build Wizard. (Same as [Build|Media Build Wizard.](#))



Compiles the project script. (Same as [Build|Compile.](#))



Runs the setup. (Same as [Build|Run Setup.](#))



Runs the setup in the Visual Debugger. (Same as [Build|Debug Setup.](#))



Launches the [Function Wizard](#) .



Prints the active text file. (Same as [File|Print.](#))



Launches your Web browser and points it to InstallShield's home page. (Same as [Help|InstallShield on the Web|InstallShield Home Page.](#))

{button ,PI("", `feedback`)} [Feedback](#)

String Table window

Displays the contents of the active string table (the string table selected in the Resources pane). Double-clicking a string resource identifier opens the Add/Modify String Entry dialog box, in which you can modify the value of that string resource.

Project Wizard-generated string resources



When entering a path as the value of a string table entry, use single backslashes, not double backslashes (as you do when directly entering a path string in the script). For example, enter C:\MyApp\Program, not C:\\MyApp\\Program.

{button ,AL(`Add/Modify String Entry dialog box;Strings window popup menu;Associate string identifiers with values
l;Resources pane (project workspace)',0,'')} See also

{button ,PI("",`feedback')} Feedback

Project Wizard-generated string resources

COMPANY_NAME

Your company's name; used by InstallationInfo() and in TARGETDIR.

COMPANY_NAME16

Used in TARGETDIR like COMPANY_NAME, but shortened to 8 characters for 16-bit setups.

ERROR_MOVEDATA

Message displayed if ComponentMoveData() returns an error.

ERROR_UNINSTSETUP

Message displayed if DeinstallStart() returns an error.

ERROR_VGARESOLUTION

Message displayed if GetExtents() returns a screen height of less than 480 pixels.

FOLDER_NAME

Default folder name shown by SdSelectFolder() program folder selection dialog.

PRODUCT_NAME

Application name; used by InstallationInfo() and SdProductName(), and in TARGETDIR.

PRODUCT_NAME16

Used in TARGETDIR like PRODUCT_NAME, but shortened to 8 characters for 16-bit setups.

PRODUCT_KEY

Application .exe; used by InstallationInfo(), RegDBSetItem(), DeinstallSetReference().

PRODUCT_VERSION

Product version number; used by InstallationInfo() to create app info registry key.

TITLE_CAPTIONBAR

Title bar of main setup window (in full window mode) and data move error dialog.

TITLE_MAIN

Title text displayed in upper left corner of main setup window; used by SetTitle().

UNINST_DISPLAY_NAME

Program name displayed in Control Panel's Add/Remove Programs dialog.

UNINST_KEY

Name of the registry's application uninstallation key; used by DeinstallStart().

{button ,PI(,"",`feedback')} Feedback

Component Properties window

Displays a list of component properties, and the values of those properties for the active component (the component selected in the Components pane). Double-clicking a property opens the corresponding property page, in which you can modify the value of that property. The following properties are available:

Description

Component description shown to your end user.

Status Text

Text shown in progress indicator during installation of component.

Display Name

Component name shown to your end user.

Visible

Whether component is visible to your end user.

Overwrite

Options for installing over already installed files.

Destination

Destination folder for component's files.

File Need

Warning to end users who deselect component.

Include in Build

Whether component is included in built disk images.

Password

Password for component.

Encryption

Whether component's files are encrypted.

CD-ROM Folder

Where component's files are placed on CD-ROM.

FTP Location

FTP location associated with component.

HTTP Location

HTTP location associated with component.

Miscellaneous

Text string associated with component (accessible from script).

Comment

Text string associated with component (not accessible from script).

Required Components

Which other components must be installed when active component is installed.

Included File Groups

File groups associated with component.

{button ,AL(`Component Properties window popup menu;Components and subcomponents !;Components pane (project workspace)',0,'')} See also

{button ,PI("",`feedback')} Feedback

File Group Properties window

Displays a list of file group properties, and the values of those properties for the active file group (the file group selected in the File Groups pane). Double-clicking a property opens the corresponding property page, in which you can modify the value of that property. The following properties are available:

Shared

Whether the file group's files are shared, that is, used by other applications.

Self Registered

Whether the file group's files are self-registering.

Potentially Locked

Whether the versions of the file group's files that are already on the end user's system might be in use during setup.

Compressed

{button ,PI("",`compress')} Visual C++ edition limitation

Whether the file group's files are compressed on the distribution media.

Operating System(s)

{button ,PI("",`OS_limits')} Visual C++ edition limitation

Operating system(s) to which the file group's files are specific.

Language(s)

Language(s) to which the file group's files are specific.

FTP Location

FTP location associated with the file group.

HTTP Location

HTTP location associated with the file group.

Miscellaneous

Text string associated with file group (accessible from script).

Comment

Text string associated with file group (not accessible from script).

{button ,AL(`File Group Properties window popup menu;Files and file groups I;File Groups pane (project workspace)',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Setup Files window

Displays information about the files in the active setup file folder (the folder selected in the Setup Files pane).



InstallShield cannot handle project file paths with more than 260 characters (including the file name). The names of the following project elements are incorporated in project paths and file names:

- n the project itself
- n file groups
- n [media names](#)
- n files in the Scripts pane
- n files inserted into the Setup Files pane

To help you keep within the 260-character limit, InstallShield warns you if you attempt to assign a name with more than 50 characters to a project, file group, or media name.



You can drag and drop files from Windows Explorer into this window. To launch Explorer, select Launch Explorer from the Tools menu.

{button ,AL(^Advanced Files icon (Setup Files pane);language icon (Setup Files pane);Language Independent icon (Setup Files pane);Setup Files pane (project workspace);Setup Files window popup menu;Splash Screen icon (Setup Files pane)',0,'')} [See also](#)

{button ,PI("",`feedback')} [Feedback](#)

Media window

This window is reserved for future use.

{button ,AL('Media pane (project workspace)',0,'')} See also

{button ,PI("",`feedback')} Feedback

Script editor window

Displays the text of the script selected in the project workspace's Scripts pane. Lets you modify the script using commands from the Edit and Tools menus.

{button ,AL(`Edit menu commands;Tools menu commands;Script editor help GS',0,`,`)} See also

{button ,PI("",`feedback')} Feedback

Setup Types window

Displays the component selections for the active setup type (the setup type selected in the Setup Types pane).



The SetupType function displays only the standard setup types Typical, Compact, and Custom, with fixed description text for each. If you want greater flexibility, call SdSetupTypeEx in your script instead.

(The Project Wizard places a call to the SetupType function in the script if you accept the default selections in the Specify Setup Types panel.)



If you try to select (as part of the active setup type) a component that you specified in the Visible property page as not visible, you get an error message. We recommend that you don't directly select invisible components, but make their selection dependent on the selection of a visible component. Otherwise, your end users can deselect all components and still see non-zero required space.

If you nonetheless want to directly select an invisible component, you can do so by first going to the Visible property page and making the component visible, then selecting it in this window, then making it invisible again.

```
{button ,AL(`Setup types;Setup Type pane (project workspace);SdSetupTypeEx I;SetupType I;Visible page  
(Component Properties sheet)',0,'')} See also
```

```
{button ,PI("",`feedback')} Feedback
```


File Group Links window

Displays a list of files stored by reference in the active Links folder (the Links folder selected in the File Groups pane).



You can drag and drop files from the Windows Explorer into this window. To launch the Explorer, select Launch Explorer from the Tools menu.



16-bit platforms such as Windows 3.1 can't read long filenames. If the platforms targeted by your setup include a 16-bit platform, don't use long filenames for your application files.



Do not place folders named TopDir or General in the root of a file group; these names are reserved in InstallShield. Placing folders named TopDir or General in the root of a file group can have undesired results, including non-installation of some files in the file group.

{button ,AL(`Organize my files into file groups I;File Groups pane (project workspace);Links icon (File Groups pane)',0,`,`')} See also

{button ,PI(`",`feedback')} Feedback

Resources window

Lists and describes types of setup resources.

String Table window popup menu

New

Opens the Add/Modify String Entry dialog box, with the string identifier initially set to NEW_STRINGn. The settings you make in this dialog box will be saved as a new string table entry when you click the OK button.

Rename

Lets you edit the identifier of the selected string resource.

Delete

Deletes the selected string resource.

Properties...

Opens the Add/Modify String Entry dialog box, in which you can view or modify the identifier, value, and comment text of the selected string resource.

{button ,AL(^Add/Modify String Entry dialog box',0,'')} See also

{button ,PI("",`feedback')} Feedback

Component Properties window popup menu

Properties

Opens the property page for the selected property.

{button ,PI("",`feedback')} Feedback

File Group Properties window popup menu

Properties

Opens the property page for the selected property.

{button ,PI("",`feedback`)}Feedback

Setup Files window popup menu

Delete

Removes the selected file from the active setup file folder.

Paste

Pastes a file (for example, a file copied from the Windows Explorer) into the active setup file folder.

Insert Files

Opens the Insert Files into <setup files folder> dialog box.

Properties...

Opens the property page of the selected file.

{button ,PI("",`feedback')} Feedback

Output window popup menu

Copy

Copies the contents of the uppermost tab to the clipboard.

Clear

Clears all text from the output window.

Hide

Disables display of the output window.

{button ,PI("",`feedback')} Feedback

Delete Project dialog box

Opens when you select Delete from the Projects window's popup menu. Lets you choose what aspects of the project you want to delete. You can choose to delete the following:

- n Only the project's icon in the Projects window. (The project's file media libraries and data are not deleted.)
- n The project's icon and file media libraries. (The project's data are not deleted.)
- n The project's icon, file media libraries, and data.

{button ,PI("", `feedback')} Feedback

General tab (New dialog box)

Displays the Blank Setup project and the Project Wizard.

Controls

Properties (button)

Opens the Project Properties sheet for the selected project.

OK (button)

If Blank Setup is selected, creates a copy of Blank Setup. If Project Wizard is selected, launches the Project Wizard.

{button ,AL(`Setup Properties sheet',0,`,`')} See also

{button ,PI(`",`feedback')} Feedback

Template tab (New dialog box)

Displays existing project templates.

Controls

Properties (button)

Opens the Project Properties sheet for the selected project template.

OK (button)

Creates a copy of the selected template.

{button ,AL(`Setup Properties sheet',0,`,`')} See also

{button ,PI(`",`feedback')} Feedback

Projects tab (New dialog box)

Displays existing projects.

Controls

Properties (button)

Opens the Project Properties sheet for the selected project.

OK (button)

Creates a copy of the selected project.

{button ,AL(`Setup Properties sheet',0,`,`')}} See also

{button ,PI("",`feedback')}} Feedback

General page (Project Properties sheet)

Displays basic facts about the active project (the project selected in the Projects window or the New dialog box). If you opened the Project Properties sheet from the Projects window, lets you choose whether to password protect the setup.

Controls

Password Protected (check box)

If checked, password protects the setup.

Set Password (button)

Opens the Set Password dialog box, in which you specify the password for the setup.

{button ,AL(' Set Password dialog box',0,','')} See also

{button ,PI('','feedback')} Feedback

Set Password dialog box

Opens when you click the Set Password button in the Project Properties sheet's General page. Lets you specify a password for your setup.

To compare this password with your end user's input during the setup, call the ComponentValidate function in your script.

Controls

New Password (edit box)

Enter a password for the setup.

Confirm Password (edit box)

Retype the password you entered in New Password. This confirms that you have entered the password you intended to enter; if your entries in New Password and Confirm Password are not the same, InstallShield warns you to re-enter the password.

{button ,PI("", `feedback`)} Feedback

Owner page (Project Properties sheet)

Lets you view information about the author of the active project (the project selected in the Projects window or the New dialog box) and the author's company. If you opened the Project Properties sheet from the Projects window, lets you modify the information in the edit boxes.

{button ,PI("",`feedback')} Feedback

Info page (Project Properties sheet)

Lets you view basic information about the active project (the project selected in the Projects window or the New dialog box). If you opened the Project Properties sheet from the Projects window, you can also modify this information.

Controls

Development Environment (list box)

Select the environment in which you developed your application.

Type (list box)

Select the type of your application.

Category (list box)

Select the category of your application.

Description (edit box)

Enter a description of the project.

Notes (edit box)

Enter notes concerning the project.

{button ,PI("",`feedback')} Feedback

Instructions page (Project Properties sheet)

Displays text to be read by anyone who might view or work with the active project (the project selected in the Projects window or the New dialog box). If you opened the Project Properties sheet from the Projects window, lets you enter and date text.

Controls

Usage Instructions/History (edit box)

Enter text you'd like read by anyone else who might view or work with your project.

Date/Time (button)

Pastes a date/time stamp into the Usage Instructions/History edit box in a new line at the end of the text.

{button ,PI("", `feedback')} Feedback

Language page (Project Properties sheet)

Lets you view the list of languages included in the active project (the project selected in the Projects window or the New dialog box). If you opened the Project Properties sheet from the Projects window, lets you add to the list of included languages.

Controls

Add (button)

Opens the Add Language dialog box, in which you can select one or more languages to add to the list of included languages.

{button ,AL(^Add Language dialog box',0,'')} See also

{button ,PI("",`feedback')} Feedback

Operating Systems page (Project Properties sheet)

Lets you view the list of operating systems included in the active project (the project selected in the Projects window or the New dialog box). If you opened the Project Properties sheet from the Projects window, lets you add to the list of included operating systems.

{button ,PI("",`OS_limits`)} Visual C++ edition limitation



16-bit platforms such as Windows 3.1 can't read long filenames. If the platforms targeted by your setup include a 16-bit platform, don't use long filenames for your application files.

Controls

Add (button)

Opens the Add Platform dialog box, in which you can select one or more operating systems to add to the list of included operating systems.

{button ,AL(`Add Platform dialog box',0,`,``)} See also

{button ,PI("",`feedback`)} Feedback

InstallShield for Visual C++ does not create setups for 16-bit operating systems.

Dialog preview (Project Wizard)

This is a preview of the dialog box you selected in the Choose Dialogs panel. This is exactly how the dialog box appears to your end user (except that you can modify elements like check box selections and text in edit boxes, as well as certain static text fields). Clicking anywhere on this preview will close it and return you to the Choose Dialogs panel.

{button ,PI("",`feedback')} Feedback

InstallShield help is attempting to send you to the InstallShield Web site (www.installshield.com).

FTP Location page (Component Properties sheet)

Lets you associate an FTP location with the active component (the component selected in the Components pane). This location can be accessed from the script using the ComponentGetData function.

{button ,AL(`ComponentGetData I',0,`,`')}} See also

{button ,PI(`",`feedback')}} Feedback

InstallShield for Visual C++ does not offer this property page.

HTTP Location page (Component Properties sheet)

Lets you associate an HTTP location with the active component (the component selected in the Components pane). This location can be accessed from the script using the ComponentGetData function.

{button ,AL(`ComponentGetData I',0,`,`')}} See also

{button ,PI("",`feedback')}} Feedback

Miscellaneous page (Component Properties sheet)

Lets you associate a text string with the active component (the component selected in the Components pane). This text can be accessed from the script using the ComponentGetData function.

Controls

Text (edit box)

Displays the text associated with the active component.

{button ,AL(`ComponentGetData I',0,`,`)} See also

{button ,PI(`",`feedback')} Feedback

FTP Location page (File Group Properties sheet)

Lets you associate an FTP location with the active file group (the file group selected in the File Groups pane).



The value set in this property page is not directly accessible from the script in this version of InstallShield. This property is included to help ensure backward compatibility in future releases of InstallShield.

HTTP Location page (File Group Properties sheet)

{button ,PI("", `feedback`)} Feedback

Lets you associate an HTTP location with the active file group (the file group selected in the File Groups pane).



The value set in this property page is not directly accessible from the script in this version of InstallShield. This property is included to help ensure backward compatibility in future releases of InstallShield.

Miscellaneous page (File Group Properties sheet)

Lets you associate a text string with the active file group (the file group selected in the File Groups pane).

Controls

Text (edit box)

Displays the text associated with the active file group.

{button ,PI("", `feedback`)} Feedback

Setup Type dialog box

Opens when you double-click a setup type in the Setup Type Properties window or select Properties from the Setup Types pane's popup menu. Lets you associate a description, a display name, and a comment with the setup type.



The SetupType function displays only the standard setup types Typical, Compact, and Custom, with fixed description text for each. If you want greater flexibility, call SdSetupTypeEx in your script instead.

(The Project Wizard puts a SetupType call in the script if you accept the default selections in the Specify Setup Types panel.)

Controls

Description (edit box)

Enter description text. This text is shown to your end users during the setup if you display the SdSetupTypeEx dialog box.

Display Name (edit box)

Enter text to be displayed to end users as the name of the setup type. If you make no entry, the setup type name (that is, the name used in the Setup Types pane and the Setup Type Properties window) is displayed to end users as the name of the setup type.

Comment (edit box)

Enter comment text. This text is not shown to your end users; it is for your use during setup creation.

{button ,AL(`SdSetupTypeEx I;SetupType I',0,'')} See also

{button ,PI("",`feedback')} Feedback

Launch Explorer command (Tools menu)

Launches the Windows Explorer.

{button ,PI("",`feedback')} Feedback

Close All command (Window menu)

Closes all windows except the Projects window, the project workspace, and the output window. (Those windows can be closed by deselecting them in the View menu.)

{button ,PI("",`feedback')} Feedback

Project Location tab (Options dialog box)

Lets you specify a folder in which new projects are placed (by default, new projects are placed in the <drive>\My Installations folder).



The locations of already existing projects are not affected by this setting.

Controls

Location of Projects (edit box)

Enter the path of the folder in which you want to place new projects.

Browse (button)

Opens the Choose Target Folder for Project(s) dialog box, in which you can select the folder in which you want to place new projects.

{button ,PI("",`feedback')} Feedback

Disk Images window

Opens when you click a disk image folder in the Media pane. Displays the contents of the folder.

{button ,PI("",`feedback')} Feedback

Folder page (Folder Properties sheet)

Lets you specify the name of the folder and (in Windows NT setups) the folder type.

Controls

Displayed Folder Name (edit box)

Lets you specify the name that the setup gives the folder when creating it on the target system. You can specify the name in any of the following ways:

- n You can enter a name in this edit box.
- n If the folder is a subfolder of the Programs folder you can enter the following text (including the angle brackets):

<SHELL_OBJECT_FOLDER>

If you do, then when the folder is created by a call to the CreateShellObjects function, the displayed folder name is set to the value of the SHELL_OBJECT_FOLDER system variable. You can set the value of SHELL_OBJECT_FOLDER with an assignment statement or with a function call such as the following:

```
SdSelectFolder( "" , "" , SHELL_OBJECT_FOLDER );
```

- n You can click the >> button to specify a string resource as the displayed name.

>> (button)

Opens the String Entry dialog box. In the String Entry dialog box, click the OK button to specify the string resource in the String Identifier list box as the folder name.

Type (option buttons)

Lets you specify the folder's type for Windows NT setups. (On other target operating systems, this setting is ignored.)

```
{button ,AL(`String Entry dialog box;CreateProgramFolder I;CreateShellObjects I;Set the name of an IDE-created program folder or group to a script-defined value;Group Properties sheet',0,`,`')}} See also
```

```
{button ,PI(``,`feedback')}} Feedback
```


Shortcut page (Shortcut Properties sheet)

Lets you set properties of the shortcut. Several of this page's properties correspond to properties on a Windows shortcut property page.

Controls



String table entries can be used only for the Shortcut Text property, by clicking the >> button. If you enter a string table ID for any other property, InstallShield interprets it as a string literal.

Location (static text field)

Displays the name of the folder the shortcut is in.

Shortcut Text (edit box)

Lets you specify the name with which the shortcut is displayed on the target system. Enter a name in this edit box or click the >> button to specify a string resource as the displayed name.

>> (button)

Opens the String Entry dialog box. In the String Entry dialog box, click the OK button to specify the string resource in the String Identifier list box as the shortcut name.

Target (edit box)

Lets you specify the fully-qualified name of the item that the shortcut points to. Together with the Parameters property, corresponds to the Target property on a Windows shortcut property page.

For information on setting this property's value equal to a system variable, click [here](#).

For information on setting this property's value from the setup script, click [here](#).

Parameters (edit box)

Lets you specify any command line parameters you want to pass to an executable specified in the Target property. Together with the Target property, corresponds to the Target property on a Windows shortcut property page.

For information on setting this property's value equal to a system variable, click [here](#).

For information on setting this property's value from the setup script, click [here](#).

Start In (edit box)

Lets you specify the folder that contains the item specified in the Target property, or a folder that contains related files. (Not applicable if the item specified in the Target property is a folder.) Corresponds to the Start In property on a Windows shortcut property page.

Shortcut Key (edit box)

Lets you specify a keyboard shortcut to quickly open or switch to the item specified in the Target property. (Not applicable if the item specified in the Target property is a folder.) Place the cursor in the edit box and press the key or key combination that you want to specify as the keyboard shortcut. Corresponds to the Shortcut Key property on a Windows shortcut property page.

Run (list box)

Lets you specify how the item is displayed when the shortcut is double-clicked. Corresponds to the Run property on a Windows shortcut property page.

Icon File (edit box)

Lets you specify the name of the file that contains the icon you want to display. (Not applicable if the item specified in the Target property is a folder.)

Icon Index (list box)

Lets you specify the icon ordinal in the Windows executable specified in the Icon File property. (Not applicable if the item specified in the Target property is a folder.) Icon ordinal numbers begin at 0, so to display the first icon in the executable file, enter 0; to display the second, enter 1, and so on. If you are not using a Windows icon, enter 0 in this parameter.

Replace existing shortcut, if found (check box)

When checked, if the shortcut already exists on the target system it is replaced.

{button ,AL(`String Entry dialog box;AddFolderIcon I;Icon Properties sheet',0,');`')};See also

{button ,PI("",`feedback')}; Feedback

Set a property's value to a system variable

Enter any one of the following in the edit box for the property:

<TARGETDIR>

<WINDIR>

<WINSYSDIR>

<PROGRAMFILES>

<COMMONFILES>

When the shortcut is created by a call to the CreateShellObjects function, the value of the property is set to the value of the corresponding system variable.

{button ,PI("",`feedback')} Feedback

Set a property's value to a string in the script

The following steps use a shortcut's Target property as an example. In this example, the value of the string is determined by the end user's input.

1. In the Shortcut property page's Target edit box, enter the following, including the angle brackets:

```
<TargetVariable>
```

2. In the setup script, set the value of a string variable equal to the desired target; for example:

```
svTarget = "C:\\\";  
SdAskDestPath( "", "", svTarget, 0 );
```

3. Set the value of <TargetVariable> equal to the value of the string variable by calling the ComponentSetTarget function:

```
ComponentSetTarget( MEDIA, "<TargetVariable>", svTarget );
```

4. Call the CreateShellObjects function to create the shortcut:

```
CreateShellObjects( "" );
```

{button ,AL(`ComponentSetTarget I;CreateShellObjects I;SdAskDestPath I,0,';')} See also

{button ,PI("",`feedback')} Feedback

File Group page (Shortcut Properties sheet)

Lets you associate a file group with the active shortcut (the shortcut selected in the Resources pane). If you associate a file group with a shortcut, the shortcut will be installed only if the file group is selected for installation. If you do not associate a file group with a shortcut, the shortcut will always be installed.

Controls

File Group Name (list box)

Lets you select a file group to associate with the active shortcut.

{button ,PI("",`feedback')} Feedback

File Group page (Icon Properties sheet)

Lets you associate a file group with the active icon (the icon selected in the Resources pane). If you associate a file group with an icon, the icon will be installed only if the file group is selected for installation. If you do not associate a file group with an icon, the icon will always be installed.

Controls

File Group Name (list box)

Lets you select a file group to associate with the active icon.

{button ,PI("",`feedback')} Feedback

Comment page (shell object properties sheet)

Lets you add, view, or modify comment text associated with the active [shell object](#) (the shell object selected in the Resources pane). This text is not included in the final file media library; it is for your reference during the setup creation process.

Controls

Comment (edit box)

Displays and lets you modify the comment text associated with the active shell object.

{button ,PI("",`feedback')} Feedback

A shortcut, folder, icon, or group.

Shortcuts (or Icons) window

Displays a list of shortcut or icon properties, and the values of those properties for the active shortcut or icon (the shortcut or icon selected in the Resources pane). Double-clicking a property opens a property page in which you can modify the value of that property. The following properties are available:

Shortcut Text (shortcuts)

The name with which the shortcut is displayed on the target system.

Target (shortcuts)

The fully-qualified name of the item that the shortcut points to.

Description (icons)

The name with which the icon is displayed on the target system.

Command Line (icons)

The fully-qualified name of the item that the icon points to, and any command line parameters you want to pass to the item.

Run in separate memory space (icons)

Whether a 16-bit application under Windows NT must run in its own memory space.

Parameters (shortcuts)

Command line parameters you want to pass to an executable specified in the Target property.

Start In (shortcuts)

The folder that contains the item specified in the Target property, or a folder that contains related files.

Working Directory (icons)

The group that contains the item specified in the Command Line property, or a group that contains related files.

Shortcut Key

Keyboard shortcut to quickly open or switch to the item specified in the Target property.

Run

How the item is displayed when the shortcut or icon is double-clicked.

Icon File

The name of the file that contains the icon graphic you want to display.

Icon Index

The icon ordinal in the Windows executable specified in the Icon File property.

Replace existing shortcut, if found

Whether the shortcut or icon is replaced if it already exists on the target system.

File Group

File group associated with the shortcut or icon.

Comment

Text string associated with the shortcut or icon (not accessible from script).

{button ,AL('Icon Properties sheet;Shortcut page (Shell Object Properties sheet)',0,'')} See also

{button ,PI('','feedback')} Feedback

Folders (or Groups) window

Displays a list of folder or group properties, and the values of those properties for the active folder or group (the folder or group selected in the Resources pane). Double-clicking a property opens a property page in which you can modify the value of that property. The following properties are available:

Displayed Folder (or Group) Name

The name that the setup gives the folder or group when creating it on the target system.

Group File Name (groups)

Name of the program group file (.grp file) that is created on the target system.

Type

The folder's or group's type for Windows NT setups.

Comment

Text string associated with folder or group (not accessible from script).

{button ,AL(`Folder page (Shell Object Properties sheet);Group Properties sheet',0,`,`')} See also

{button ,PI(`",`feedback')} Feedback

Icon page (Icon Properties sheet)

Lets you set properties of the icon. Several of this page's properties correspond to properties on a Windows program item property page.

Controls



String table entries can be used only for the Description property, by clicking the >> button. If you enter a string table ID for any other property, InstallShield interprets it as a string literal.

Location (static text field)

Displays the name of the group the icon is in.

Description (edit box)

Lets you specify the name with which the icon is displayed on the target system. Enter a name in this edit box or click the >> button to specify a string resource as the displayed name.

>> (button)

Opens the String Entry dialog box. In the String Entry dialog box, click the OK button to specify the string resource in the String Identifier list box as the icon name.

Command Line (edit box)

Lets you specify the fully-qualified name of the item that the icon points to and any command line parameters you want to pass to the item. Corresponds to the Command Line property on a Windows program item property page.

For information on setting this property's value equal to a system variable, click [here](#).

For information on setting this property's value equal to a string variable in the script, click [here](#).

Run in separate memory space (check box)

Applies to 16-bit applications under Windows NT only. Tells the operating system that the 16-bit application must run in its own memory space. Corresponds to the "Run in separate memory space" property on a Windows program item property page.

Working Directory (edit box)

Lets you specify the group that contains the item specified in the Command Line property, or a group that contains related files. (Not applicable if the item specified in the Target property is a group.) Corresponds to the Working Directory property on a Windows program item property page.

For information on setting this property's value equal to a system variable, click [here](#).

For information on setting this property's value equal to a string variable in the script, click [here](#).

Shortcut Key (edit box)

Lets you specify a keyboard shortcut to quickly open or switch to the icon specified in the Command Line property. (Not applicable if the item specified in the Command Line property is a group.) Place the cursor in the edit box and press the key or key combination that you want to specify as the keyboard shortcut. Corresponds to the Shortcut Key property on a Windows program item property page.

Run (list box)

Lets you specify how the item is displayed when the icon is double-clicked. Corresponds to the Run Minimized property on a Windows program item property page.

Icon File (edit box)

Lets you specify the name of the file that contains the icon you want to display. (Not applicable if the item specified in the Command Line property is a group.)

Icon Index (list box)

Lets you specify the icon ordinal in the Windows executable specified in the Icon File property. (Not applicable if the item specified in the Command Line property is a group.) Icon ordinal numbers begin at 0, so to display the first icon in the executable file, enter 0; to display the second, enter 1, and so on. If you are not using a Windows icon, enter 0 in this parameter.

Replace existing icon, if found (check box)

When checked, if the icon already exists on the target system it is replaced.

{button ,AL(`String Entry dialog box;AddFolderIcon I;Shell Objects - Shortcuts window',0,`,`')}} See also
{button ,PI("",`feedback')}} Feedback

Group page (Group Properties sheet)

Lets you specify the name of the group and (in Windows NT setups) the folder type.

Controls



String table entries can be used only for the Displayed Group Name property, by clicking the >> button. If you enter a string table ID for any other property, InstallShield interprets it as a string literal.

Displayed Group Name (edit box)

Lets you specify the name that the setup gives the group when creating it on the target system. You can specify the name in any of the following ways:

- n You can enter a name in the edit box.
- n You can enter the following:

<SHELL_OBJECT_FOLDER>

If you do, then when the group is created by a call to the CreateShellObjects function, the displayed group name is set to the value of the SHELL_OBJECT_FOLDER system variable. You can set the value of SHELL_OBJECT_FOLDER with an assignment statement or with a function call such as the following:

```
SdSelectFolder( "" , "" , SHELL_OBJECT_FOLDER );
```

- n You can click the >> button to specify a string resource as the displayed name.

>> (button)

Opens the String Entry dialog box. In the String Entry dialog box, click the OK button to specify the string resource in the String Identifier list box as the group name.

Type (option buttons)

Lets you specify the group's type for Windows NT setups. (On other target operating systems, this setting is ignored.)

Group File Name (edit box)

Lets you specify the name of the program group file (.grp file) that is created on the target system. Applies to 16-bit target platforms only, not to Windows NT 3.51.

```
{button ,AL(`String Entry dialog box;ProgDefGroupType I;CreateProgramFolder I;CreateShellObjects I;Set the name of an IDE-created program folder or group to a script-defined value;Folder page (Shell Object Properties sheet)',0,`')} See also
```

```
{button ,PI(`",`feedback')} Feedback
```

Registry Entries window

Lets you add or modify registry keys and values in a registry set. You can create registry sets in the Resources pane of the project workspace.

When adding subkeys, use backslashes to separate the different levels. For example, to add the key Key2 as a subkey of Key1 under HKEY_CLASSES_ROOT, create a key under HKEY_CLASSES_ROOT and name the key Key1\Key2.



If you created your setup project using the Project Wizard, your script contains function calls that create the primary setup-related registry keys: the application information key, the application uninstallation key, and the App Paths key. In this case, there is no need to create the keys in a registry set; doing so will yield undesired results.



If you add only Key1\Key2\Key3 to a registry set, only Key3 is logged for uninstallation, while Key2 and Key1 are not. To have all these keys logged for uninstallation, create Key1 separately; then when Key1 is uninstalled, all its subkeys (Key2 and Key3) are also uninstalled.

{button ,AL(`Registry Entries window popup menu;Resources pane (project workspace)',0,`,`)} See also
{button ,PI(`",`feedback`)} Feedback

Registry Entries window popup menu

Modify

Lets you edit the name of the selected key or the name and value of the selected value.



You cannot use a string table entry as a key name.

New

Opens a submenu with the following commands:

Key

Creates a new key under the selected root key or under the root key of the selected key.

String Value

Creates a new string value under the selected key.

Binary Value

Creates a new binary value under the selected key.

DWORD Value

Creates a new DWORD value under the selected key.

Delete

Deletes the selected key or value.

{button ,AL(`Registry Entries window',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Edit String dialog box

Lets you modify string values of registry keys.

Controls

Value Name (edit box)

Enter a name for the selected string value. (If the default value of a key is selected, this field is uneditable.)



You cannot specify a value name that contains an equals sign (=).

Value Data (edit box)

Lets you specify the value of the string. You can specify the name in any of the following ways:

- n You can enter string text in this edit box.
- n You can enter any one of the following:

<TARGETDIR>
<WINDIR>
<WINSYSDIR>
<PROGRAMFILES>
<COMMONFILES>

If you do, then when the registry key is created by a call to the CreateRegistrySet function, the string value is set to the value of the corresponding system variable.

- n You can click the >> button to specify a string resource as the string value.

>> (button)

Opens the String Entry dialog box. In the String Entry dialog box, click the OK button to specify the string resource in the String Identifier list box as the string value.



You cannot use multiple string table entries in any single key value, nor can you use a string table entry in combination with any hard coded strings, script defined folders, system variables, etc.

The following is an example of the only valid use of a string table entry as a key value:

```
MyStringValue="STRING_TABLE_ID"
```

The following are examples of invalid uses:

- n MyStringValue="STRING_TABLE_ID1\STRING_TABLE_ID2"
- n MyStringValue="C:\MyApp\STRING_TABLE_ID"
- n MyStringValue="<TARGETDIR>\STRING_TABLE_ID"
- n MyStringValue="<MyScriptDefinedFolder>\STRING_TABLE_ID"

{button ,AL(`CreateRegistrySet I;Registry Entries window;String Entry dialog box',0,'')} See also

{button ,PI("",`feedback')} Feedback

Edit Binary Value dialog box

Lets you modify binary values of registry keys.

Controls

Value Name (edit box)

Enter a name for the selected binary value.



You cannot specify a value name that contains an equals sign (=).

Value Data (edit box)

Enter the hexadecimal representation of the binary value.



This dialog box only accepts values with an even number of hexadecimal digits (since two hexadecimal digits represents one byte of data). If you enter a value with an odd number of hexadecimal digits, the OK button is disabled.



You cannot use a string table entry as a binary key value.

{button ,AL('Registry Entries window',0,'')} See also

{button ,PI('','feedback')} Feedback

Edit DWORD Value dialog box

Lets you modify DWORD values of registry keys.

Controls

Value Name (edit box)

Enter a name for the selected DWORD value.



You cannot specify a value name that contains an equals sign (=).

Value Data (edit box)

Enter a DWORD value.



You cannot use a string table entry as a DWORD key value.

Base (options buttons)

Select whether the value entered in the Value Data edit box is interpreted as a decimal or a hexadecimal value.
Convert the value to another base by selecting the button for that base.

{button ,AL(`Registry Entries window',0,`,`')} See also

{button ,PI("",`feedback')} Feedback

Unavailable in Visual C++ edition

This feature is unavailable in InstallShield for Microsoft Visual C++.

{button ,PI("",`feedback')} Feedback

General page (Registry Entry Properties sheet)

The Registry Entry Properties sheet opens when you right-click a registry set in the Resources pane and select Modify from the popup menu.

The General page lets you add, view, or modify comment text associated with the active registry set (the registry set selected in the Resources pane). This text is not included in the final file media library; it is for your reference during the setup creation process.

Controls

Comment (edit box)

Displays and lets you modify the comment text associated with the active registry set.

{button ,PI("",`feedback')} Feedback

File Group page (Registry Entry Properties sheet)

The Registry Entry Properties sheet opens when you right-click a registry set in the Resources pane and select Modify from the popup menu.

The File Group page lets you associate a file group with the active registry set (the registry set selected in the Resources pane). The keys and values in the registry set are created on the target system if the associated file group is installed and the CreateRegistrySet function is called. If no file group is associated with the registry set, its keys and values are created on the target system when the CreateRegistrySet function is called.

Controls

File Groups (list box)

Select the file group to associate with the active registry set.

{button ,AL(^CreateRegistrySet I;Resources pane (project workspace)',0,'')} See also

{button ,PI("",`feedback')} Feedback

We welcome your suggestions for improving InstallShield help. To share your ideas, please visit <http://www.installshield.com/talk/> or email doc@installshield.com.

InstallShield for Visual C++ does not have the Component Wizard.

InstallShield for Visual C++ does not let you create more than five components or create subcomponents.

