Setup and Deployment Projects

Article • 11/01/20125 minutes to read

Visual Studio provides templates for four types of deployment projects: Merge Module Project, Setup Project, Web Setup Project, and CAB Project. In addition, a Setup wizard helps you create deployment projects. You can see the templates and the wizard in the **New Project** dialog box. Expand the **Other Project Types** node, select the **Setup and Deployment Projects** node, and then click **Visual Studio Installer**.

You can also enable InstallShield Limited Edition to create, build, and package applications and services for Windows. To begin, in the **New Project** dialog box, expand the **Other Project Types** node, select the **Setup and Deployment Projects** node, and then click **InstallShield LE**.

Template Overview

The following guidelines help you choose the correct type of deployment project for your project.

Project Type	Purpose
InstallShield	Creates, builds, and packages applications and services for Windows by using InstallShield 2010 Limited Edition.
Merge Module Project	Packages components that can be shared by multiple Windows-based applications. Merge Module projects let you package files or components into a single module, which makes the module easier to share. The resulting .msm files can be included in any other deployment project.
Setup Project	Builds an installer for a Windows-based application. The files for a Setup project are installed into the Program Files directory on end-user computers.
Web Setup Project	Builds an installer for a Web application. The files for a Web Setup Projects are installed into a Virtual Root directory on Web servers.
CAB Project	Creates a CAB file for downloading to an earlier Web browser. You can use CAB projects to package ActiveX components that can be downloaded from a Web server to a Web browser.

⚠ **Warning**

After you create a project, you cannot change its type from Setup Project to Web Setup project. If you create a Setup project and later decide to deploy it to a Web, you must create a new Web Setup project.

Setup Projects

Setup projects allow you to create installers in order to distribute an application through CD or network file share. The resulting Windows Installer (.msi) file contains the application, any dependent files, information about the application such as registry entries, and instructions for installation. When the .msi file is distributed and run on another computer, you can be assured that everything necessary for installation is included; if for any reason the installation fails (for example, the target computer does not have the required operating system version), the installation will be rolled back and the computer returned to its preinstallation state.

There are two types of setup projects in Visual Studio: **Setup** projects and **Web Setup** projects. The distinction between **Setup** and **Web Setup** projects is where the installer will be deployed: **Setup** projects will install files into the file system of a target computer; **Web Setup** projects install files into a virtual directory of a Web server.

In addition, a **Setup Wizard** is available to simplify the process of creating a **Setup** or **Web Setup** project.

Web Setup Projects

In addition to creating Windows Installers for distribution via traditional media, the deployment tools in Visual Studio also support deployment to a Web server. Using deployment to install files on a Web server provides an advantage over simply copying files, in that deployment handles any issues with registration and configuration automatically.

In addition, installers for Windows applications can be deployed to a Web server so that users can later download and run them from a Web site.

To deploy a Web application to a Web server, you create a **Web Setup** project, build it, copy it to the Web server computer, and run the installer to install the application on the server using the settings defined in your **Web Setup** project.

ⓘ **Note**

In order to deploy to a Web server, you must have administrative access privileges for that computer. For more information, see Web Pages and Projects .

In addition, the behavior of some administrative settings with regard to deployment may not be obvious. If the Internet Information Services (IIS) **Write** property for a virtual directory is disabled, files will still be deployed to that directory; the **Write** property only controls the ability for users to upload files.

To deploy an application for download from a Web server, you create a **Web Setup** project and add the project output group for the application to the **Web Setup** project in the **File System Editor**. After building the installer, you copy it to the Web server computer, where it can then be downloaded via a Web browser.

Note

When an installer is made available for download, there is no guarantee that the Windows Installer run-time files will be available on the user's computer. You should always make sure that the run-time files are also available for download by specifying a location in the **Web Bootstrapper Settings** dialog box. For more information, see Web Bootstrapper Settings Dialog Box.

Cab File Projects

Cab projects allow you to create a .Cab file to package ActiveX controls that can be downloaded from a Web server to a Web browser.

Unlike the other deployment project types, there are no editors provided for working with Cab projects. Files and project outputs can be added to a Cab project in **Solution Explorer**, and properties can be set in the **Properties** window or in the **Project Property** pages.

Properties of Cab projects allow you to specify a level of compression, implement Authenticode signing, set the display name and version information, and specify the location of dependent files on the Web.

Note

Dependencies are not calculated for Cab projects; you must determine any dependencies and reference them.

Merge Module Projects

Merge Module projects allow you to create reusable setup components. Much the same as dynamic-link libraries allow you to share code between applications, merge modules allow you to share setup code between Windows Installers.

A merge module (.msm file) is a single package that contains all files, resources, registry entries, and setup logic necessary to install a component. Merge modules cannot be installed alone, but must be used within the context of a Windows Installer (.msi) file. Merge modules allow you to capture all of the dependencies for a particular component, ensuring that the correct versions are installed. Once you have distributed a merge module, it should never be modified; instead you should create a new merge module for each successive version of your component.

However, we recommend that you distribute dependencies and prerequisites by using a chainer or bootstrapper. For more information about bootstrappers, see Application Deployment Prerequisites.

See Also

Tasks

How to: Add Items to a Deployment Project

How to: Create or Add Deployment Projects

Reference

General, Projects and Solutions, Options Dialog Box

Concepts

What's New in Deployment