

Installing and Maintaining Programs

- Managing Application Virtualization and Run Levels **311**
- Installing Programs: The Essentials **318**
- Deploying Applications Through Group Policy **322**
- Configuring Program Compatibility **324**
- Managing Installed and Running Programs **328**

Administrators and support staff often install and configure applications that are used on desktop computers. You need to install and configure applications before deploying new computers, install new applications on computers when the programs are requested, and update applications when new versions become available. Also, as users install additional applications, you might be called on to help troubleshoot installation problems or to help uninstall programs. Most program installation problems are fairly easy to solve if you know what to look for. Other problems are fairly difficult to resolve and require more work than you might expect. In this chapter, you'll learn how User Account Control (UAC) affects how you install and run applications and about techniques for installing, uninstalling, and maintaining programs.

Managing Application Virtualization and Run Levels

User Account Control (UAC) changes the way that applications are installed and run, where applications write data, and what permissions applications have. In this section, I'll look at how UAC affects application installation, from application security tokens to file and registry virtualization to run levels. This information is essential when you are installing and maintaining applications on Windows 7.

Application Access Tokens and Location Virtualization

All applications used with Windows 7 are divided into two general categories:

- **UAC-compliant** Any application written specifically for Windows Vista or Windows 7 is considered a compliant application. Applications certified as complying with the Windows 7 architecture have the UAC-compliant logo.
- **Legacy** Any application written for Windows XP or an earlier version of Windows is considered a legacy application.

The distinction between UAC-compliant applications and legacy applications is important because of the architectural changes required to support UAC. UAC-compliant applications use UAC to reduce the attack surface of the operating system. They do this by preventing unauthorized programs from installing or running without the user's consent and by restricting the default privileges granted to applications. These measures make it harder for malicious software to take over a computer.

NOTE The Windows 7 component responsible for UAC is the Application Information service. This service facilitates the running of interactive applications with an "administrator" access token. You can see the difference between the administrator user and standard user access tokens by opening two Command Prompt windows, running one with elevation (right-click, and then click Run As Administrator), and the other as a standard user. In each window, type `whoami /all` and compare the results. Both access tokens have the same security identifiers (SIDs), but the elevated, administrator user access token will have more privileges than the standard user access token.

All applications that run on Windows 7 derive their security context from the current user's access token. By default, UAC turns all users into standard users even if they are members of the Administrators group. If an administrator user consents to the use of her administrator privileges, a new access token is created for the user. It contains all the user's privileges, and this access token—rather than the user's standard access token—is used to start an application or process.

In Windows 7, most applications can run using a standard user access token. Whether applications need to run with standard or administrator privileges depends on the actions the application performs. Applications that require administrator privileges, referred to as *administrator user applications*, differ from applications that require standard user privileges, referred to as *standard user applications*, in the following ways:

- Administrator user applications require elevated privileges to run and perform core tasks. Once started in elevated mode, an application with a user's administrator access token can perform tasks that require administrator privileges and can also write to system locations of the registry and the file system.

- Standard user applications do not require elevated privileges to run or to perform core tasks. Once started in standard user mode, an application with a user's standard access token must request elevated privileges to perform administration tasks. For all other tasks, the application should not run using elevated privileges. Further, the application should write data only to nonsystem locations of the registry and the file system.

Applications not written for Windows 7 run with a user's standard access token by default. To support the UAC architecture, these applications run in a special compatibility mode and use file system and registry virtualization to provide "virtualized" views of file and registry locations. When an application attempts to write to a system location, Windows 7 gives the application a private copy of the file or registry value. Any changes are then written to the private copy, and this private copy is then stored in the user's profile data. If the application attempts to read or write to this system location again, it is given the private copy from the user's profile to work with. By default, if an error occurs when the application is working with virtualized data, the error notification and logging information show the virtualized location rather than the actual location that the application was trying to work with.

Application Integrity and Run Levels

The focus on standard user and administrator privileges also changes the general permissions required to install and run applications. In Windows XP and earlier versions of Windows, the Power Users group gave users specific administrator privileges to perform basic system tasks when installing and running applications. Applications written for Windows 7 do not require the use of the Power Users group. Windows 7 maintains it only for legacy application compatibility.

As part of UAC, Windows 7 by default detects application installations and prompts users for elevation to continue the installation. Installation packages for UAC-compliant applications use application manifests that contain run-level designations to help track required privileges. Application manifests define the application's privileges as one of the following:

- **RunAsInvoker** Run the application with the same privileges as the user. Any user can run the application. For a standard user or a user who is a member of the Administrators group, the application runs with a standard access token. The application runs with higher privileges only if the parent process from which it is started has an administrator access token. For example, if you open an elevated Command Prompt window and then launch an application from this window, the application runs with an administrator access token.
- **RunAsHighest** Run the application with the highest privileges of the user. The application can be run by both administrator users and standard users. The tasks the application can perform depend on the user's privileges. For a standard user, the application runs with a standard access token. For a user who is a member of a group with additional privileges, such as the Backup

Operators, Server Operators, or Account Operators group, the application runs with a partial administrator access token that contains only the privileges the user has been granted. For a user who is a member of the Administrators group, the application runs with a full administrator access token.

- **RunAsAdmin** Run the application with administrator privileges. Only administrators can run the application. For a standard user or a user who is a member of a group with additional privileges, the application runs only if the user can be prompted for credentials required to run in elevated mode or if the application is started from an elevated process, such as an elevated Command Prompt window. For a user who is a member of the Administrators group, the application runs with an administrator access token.

To protect application processes, Windows 7 labels them with integrity levels ranging from high to low. Applications that modify system data, such as Disk Management, are considered high integrity. Applications performing tasks that could compromise the operating system, such as Windows Internet Explorer 8 in Windows 7, are considered low integrity. Applications with lower integrity levels cannot modify data in applications with higher integrity levels.

Windows 7 identifies the publisher of any application that attempts to run with an administrator's full access token. Then, depending on that publisher, Windows 7 marks the application as belonging to one of the following three categories:

- Windows Vista / Windows 7
- Publisher verified (signed)
- Publisher not verified (unsigned)

To help you quickly identify the potential security risk of installing or running the application, a color-coded elevation prompt displays a particular message depending on the category to which the application belongs:

- If the application is from a blocked publisher or is blocked by Group Policy, the elevation prompt has a red background and displays the message "The application is blocked from running."
- If the application is administrative (such as Computer Management), the elevation prompt has a blue-green background and displays the message "Windows needs your permission to continue."
- If the application has been signed by Authenticode and is trusted by the local computer, the elevation prompt has a gray background and displays the message "A program needs your permission to continue."
- If the application is unsigned (or is signed but not yet trusted), the elevation prompt has a yellow background and red shield icon and displays the message "An unidentified program wants access to your computer."

Prompting on the secure desktop can be used to further secure the elevation process. The secure desktop safeguards the elevation process by preventing spoofing of the elevation prompt. The secure desktop is enabled by default in Group

Policy, as discussed in the section “Optimizing User Account Control and Admin Approval Mode” in Chapter 5.

Setting Run Levels

By default, only applications running with a user’s administrator access token run in elevated mode. Sometimes, you’ll want an application running with a user’s standard access token to be in elevated mode. For example, you might want to start the Command Prompt window in elevated mode so that you can perform administration tasks.

In addition to application manifests (discussed in the previous section), Windows 7 provides two different ways to set the run level for applications:

- Run an application once as an administrator.
- Always run an application as an administrator.

To run an application once as an administrator, right-click the application’s shortcut or menu item, and then click Run As Administrator. If you are using a standard account and prompting is enabled, you are prompted for consent before the application is started. If you are using a standard user account and prompting is disabled, the application will fail to run. If you are using an administrator account and prompting for consent is enabled, you are prompted for consent before the application is started.

Windows 7 also enables you to mark an application so that it always runs with administrator privileges. This approach is useful for resolving compatibility issues with legacy applications that require administrator privileges. It is also useful for UAC-compliant applications that normally run in standard mode but that you use to perform administration tasks. As examples, consider the following:

- A standard application written for Windows 7 is routinely run in elevated mode and used for administration tasks. To eliminate the need to right-click the application shortcut and choose Run As Administrator before running the application, you can mark it to always run as an administrator.
- An application written for Windows XP or an earlier version of Windows requires administrator privileges. Because this program is configured to use standard mode by default under Windows 7, the program isn’t running properly and is generating numerous errors. To resolve the compatibility problem, you could create an application compatibility shim using the Windows Application Compatibility Toolkit (ACT) version 5.5 or later. As a temporary solution, you can mark the application to always run as an administrator.

NOTE You cannot mark system applications or processes to always run with administrator privileges. Only nonsystem applications and processes can be marked to always run at this level.

REAL WORLD The Windows Application Compatibility Toolkit (ACT) is a solution for administrators that requires no reprogramming of an application. ACT can help you resolve common compatibility problems. For example, some programs run only on a specific operating system or when the user is an administrator. Using ACT, you can create a shim that responds to the application inquiry about the operating system or user level with a True statement, which allows the application to run. ACT also can help you create more in-depth solutions for applications that try to write to protected areas of the operating system or use elevated privileges when they don't need to. ACT can be downloaded from the Microsoft Download Center (<http://download.microsoft.com>).

You can mark an application to always run as an administrator by following these steps:

1. On the Start menu, locate the program that you want to always run as an administrator.
2. Right-click the application's shortcut, and then click Properties.
3. In the Properties dialog box, click the Compatibility tab, shown in Figure 9-1.

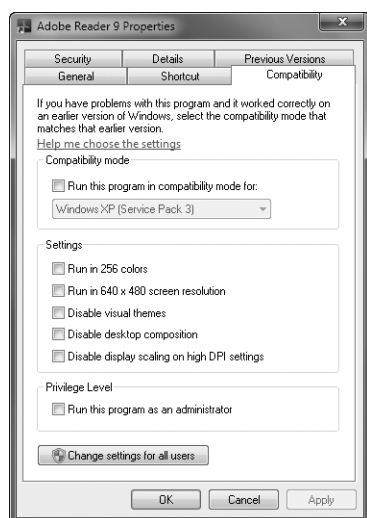


FIGURE 9-1 Access the Compatibility tab.

4. Do one of the following:
 - To apply the setting to the currently logged-on user, select the Run This Program As An Administrator check box, and then click OK.
 - To apply the setting to all users on the computer and regardless of which shortcut is used to start the application, click Change Setting For All Users to display the Properties dialog box for the application's .exe file, select the Run This Program As An Administrator check box, and then click OK twice.

NOTE If the Run This Program As An Administrator option is unavailable, it means that the application is blocked from always running at an elevated level, the application does not require administrator credentials to run, or you are not logged on as an administrator.

The application will now always run using an administrator access token. Keep in mind that if you are using a standard account and prompting is disabled, the application will fail to run.

Optimizing Virtualization and Installation Prompting for Elevation

With regard to applications, two areas of User Account Control can be customized:

- Automatic installation detection and prompting
- Virtualization of write failures

In Group Policy, you can configure these features by using the Administrative Templates policies for Computer Configuration under Windows Settings\Security Settings\Local Policies\Security Options. The security settings are as follows:

- **User Account Control: Detect Application Installations And Prompt For Elevation** Determines whether Windows 7 automatically detects application installation and prompts for elevation or consent. (This setting is enabled by default in Windows 7.) If you disable this setting, users are not prompted, in which case, the users will not be able to elevate permissions by supplying administrator credentials.
- **User Account Control: Virtualize File And Registry Write Failures To Per-User Locations** Determines whether file and registry virtualization is on or off. Because this setting is enabled by default, error notifications and error logging related to virtualized files and registry values are written to the virtualized location rather than the actual location to which the application was trying to write. If you disable this setting, the application will silently fail when trying to write to protected folders or protected areas of the registry.

In a domain environment, you can use Active Directory–based Group Policy to apply the security configuration you want to a particular set of computers. You can also configure these settings on a per-computer basis by using local security policy. To do this, follow these steps:

1. Click Start, point to All Programs, Administrative Tools, and then click Local Security Policy. This starts the Local Security Policy console.
2. In the console tree, under Security Settings, expand Local Policies, and then select Security Options.
3. Double-click the setting you want to work with, make any necessary changes, and then click OK.

Installing Programs: The Essentials

Program installation is fairly straightforward. Not so straightforward are troubleshooting the many things that can go wrong and fixing problems. To solve problems that might occur, you first need to understand the installation process. In many cases, the typical installation process starts when Autorun is triggered. Autorun in turn invokes a setup program. Once the setup program starts, the installation process can begin. Part of the installation process involves checking the user's credentials to ensure that he or she has the appropriate privileges to install the program and prompting for consent if the user doesn't. As part of installing a program, you might also need to make the program available to all or only some users on a computer.

Occasionally, Windows might not be successful in detecting the required installation permissions. This can occur if the installation manifest for the program has an embedded `RequestedExecutionLevel` setting that has a value set as `RequireAdministrator`. Because the `RequestedExecutionLevel` setting overrides what the installer detects in Windows, the installation process fails any time you run the installer with standard user permissions. To solve this problem, back out of the failed installation by exiting, canceling the installation, or taking another appropriate action. Next, locate the executable file for the installer. Right-click this file, and then click **Run As Administrator** to restart the installation process with administrator privileges.

Additionally, it is important to understand that in Windows 7 and Windows Server 2008 Release 2, Application Control policies replace Software Restriction policies. Software Restriction policies control the applications that users can install and run on Windows 2000, Windows XP, and Windows Vista. Application Control policies control the applications that users can install and run on Windows 7 and Windows Server 2008 Release 2. Keep the following in mind:

- When you are editing a Group Policy object (GPO), you can create and manage Software Restriction policies by using settings for computers under `Computer Configuration\Policies\Windows Settings\Security Settings\Software Restriction Policies` and settings for users under `User Configuration\Policies\Windows Settings\Security Settings\Software Restriction Policies`. Enforcement settings control how restrictions are applied. Designated file types determine what is and what is not considered an executable program.
- When you are editing a GPO, you can create and manage Application Control policies by using settings for computers under `Computer Configuration\Policies\Windows Settings\Security Settings\Application Control Policies`. You can now create separate rules for executable files, Windows installer files, and script files. Rules can be applied by publisher, file path, or file hash. A publisher rule gives you the most flexibility, enabling you to specify which products and versions to allow. For example, you could allow Microsoft Word 2003 or later.

Working with Autorun

When you insert an application CD or DVD into a CD or DVD drive, Windows 7 checks for a file named `Autorun.inf`. If present, `Autorun.inf` specifies the action that the operating system should take and might also define other installation parameters. `Autorun.inf` is a text-based file that can be opened in any standard text editor. If you were to examine the contents of one, you'd see something similar to the following code:

```
[autorun]
OPEN=SETUP.EXE AUTORUN=1
ICON=SETUP.EXE,4
SHELL=OPEN
DisplayName=Microsoft Digital Image Suite 9
ShortName=PIS
PISETUP=PIP\pisetup.exe
```

This `Autorun.inf` file opens a file named `Setup.exe` when the CD or DVD is inserted into the CD or DVD drive. Because `Setup.exe` is an actual program, this program is invoked. The `Autorun.inf` file also specifies an icon to use, the status of the shell, the program display name, the program's short name, and an additional parameter, which in this case is the location of another setup program to run.

The file that `Autorun.inf` specifies to open won't always be a program. Consider the following example:

```
[autorun]
OPEN=Autorun\ShellExec default.htm
```

This `Autorun.inf` file executes via the shell and opens a file named `Default.htm` in the computer's Web browser. It's important to note that even in this case, the document opened in the Web browser contains links that point to a setup program.

TIP With an application CD or DVD in a drive, you can restart the Autorun process at any time. Simply open and then close the drive bay.

Application Setup and Compatibility

Most applications have a setup program that uses InstallShield, Wise Install, or Microsoft Windows Installer. When you start the setup program, the installer helps track the installation process and should also make it possible to easily uninstall the program when you need to. If you are installing an older application, the setup program might use an older version of one of these installers, and this might mean the uninstall process won't completely uninstall the program.

Even if you are absolutely certain that a program has a current installer, you should consider the possibility that you will need to recover the system if something goes wrong with the installation. To help ensure that you can recover your system, check that System Restore is enabled for the drive on which you are installing

the program so that System Restore can create an automatic checkpoint before installing the program.

While the installers for most current programs automatically trigger the creation of a restore point before making any changes to a computer, the installers for older programs might not. You can manually create a restore point as discussed in Chapter 17, “Handling Maintenance and Support Tasks.” Then, if you run into problems, you can try to uninstall the program or use System Restore to recover the system to the state it was in prior to the program’s installation.

Before installing any application, you should check to see whether it is compatible with Windows 7. To determine compatibility, you can do the following:

- Check the software packaging, which should specify whether the program is compatible. Look for the Windows 7 logo.
- Check the software developer’s Web site for a list of compatible operating systems.

NOTE As part of the compatibility check, look for updates or patches for the program. If any are available, install them after installing the program.

Windows 7 attempts to recognize potential compatibility problems before you install applications. If it detects one, you might see a Program Compatibility Assistant dialog box after you start a program’s installer. Often, this dialog box contains information about the known compatibility issues with the program, and in many cases it displays a possible solution. For example, you might be advised to install the latest service pack for the program before running the program on the computer. In some cases, the Program Compatibility Assistant might display the message “This program is blocked due to compatibility issues.” Here, the program is blocked because it causes a known stability issue with Windows, and you can’t create an immediate fix to work around the problem. Your only options are to click the Check For Solutions Online button or click Cancel. If you check for solutions online, the typical solution requires you to purchase an updated version of the program. If you cancel, you stop the installation process without checking for possible solutions.

If the installation continues but fails for any reason before it is fully complete (or to properly notify the operating system regarding completion), you’ll also see a Program Compatibility Assistant dialog box. In this case, if the program installed correctly, click This Program Installed Correctly. If the program didn’t install correctly, click Reinstall Using Recommended Settings to allow the Program Compatibility Assistant to apply one or more compatibility fixes, and then try again to run the installer.

When you start programs, Windows 7 uses the Program Compatibility Assistant to automatically make changes for known compatibility issues as well. If the Program Compatibility Assistant detects a known compatibility issue when you run an application, it notifies you about the problem and provides possible solutions for resolving the problem automatically. You can then allow the Program Compatibility

Assistant to reconfigure the application for you, or you can manually configure compatibility as discussed in the section “Configuring Program Compatibility” later in this chapter.

For legacy applications, you can also use the Compatibility Administrator (Compatadmin.exe), provided in the Windows Application Compatibility Toolkit, to create an application manifest that sets the application’s run level. The Compatibility Administrator can also help identify other types of compatibility issues with legacy applications. The Windows Application Compatibility Toolkit (ACT) can be downloaded from the Microsoft Download Center (<http://download.microsoft.com>).

Making Programs Available to All or Selected Users

Usually when you install a program, the program is available to all users on a computer. This occurs because the program’s shortcuts are placed in the Start Menu\Programs folder (%SystemDrive%\ProgramData\Microsoft\Windows\Start Menu\Programs) for all users so that any user who logs on to the system has access to the program. Some programs prompt you during installation to choose whether you want to install the program for all users or only for the currently logged-on user. Other programs simply install themselves only for the current user.

If setup installs a program so that it is available only to the currently logged-on user and you want other users to have access to the program, you need to take one of the following actions:

- Log on to the computer with each user account that should have access to the program, and then rerun setup to make the program available to these users. You also need to run setup again when a new user account is added to the computer and that user needs access to the program.
- For programs that don’t require per-user settings to be added to the registry before running, you can in some cases make the program available to all users on a computer by adding the appropriate shortcuts to the Start Menu\Programs folder for all users. Copy or move the program shortcuts from the currently logged-on user’s profile to the Start Menu\Programs folder for all users.

If you want to make a program available to all users on a computer, you can copy or move a program’s shortcuts by completing the following steps:

1. Right-click the Start button, and then click Open Windows Explorer. In Windows Explorer, navigate to the currently logged on user’s Programs folder. This is a hidden folder under %UserProfile%\AppData\Roaming\Microsoft\Windows\Start Menu.
2. In the Programs folder, right-click the folder for the program group or the shortcut you want to work with, and then click Copy or Cut on the shortcut menu.

3. Next, navigate to the all-users Start Menu\Programs folder. This hidden folder is under %SystemDrive%\ProgramData\Microsoft\Windows\Start Menu.
4. In the Programs folder, right-click an open space, and then click Paste. The program group or shortcut should now be available to all users of the computer.

NOTE In the %SystemDrive%\Users folder, you'll find a folder called All Users. If you are aware of this folder, you might wonder why you didn't copy the program's shortcut for all users to a subfolder of this folder. Well, the reason is that %SystemDrive%\Users\All Users is a symbolic link to %SystemDrive%\ProgramData. A symbolic link is a pointer to where a folder actually exists. When you are working with the command prompt (Cmd.exe), you can view symbolic links and reparse points (junctions) in the current directory by entering `dir /al`.

If you want to make a program available only to the currently logged-on user rather than all users on a computer, you can move a program's shortcuts by completing the following steps:

1. Right-click the Start button, and then click Open Windows Explorer. In Windows Explorer, navigate to the all-users Start Menu folder. This hidden folder is under %SystemDrive%\ProgramData\Microsoft\Windows\Start Menu.
2. In the Programs folder, right-click the folder for a program group or the program shortcut that you want to work with, and then click Cut.
3. In Windows Explorer, navigate to the currently logged-on user's Programs folder. This is a hidden folder under %UserProfile%\AppData\Roaming\Microsoft\Windows\Start Menu.
4. In the Programs folder, right-click an open space, and then click Paste. The program group or shortcut should now be available only to the currently logged-on user.

NOTE Moving a program group or shortcut hides the fact that the program is available on the computer—it doesn't prevent other users from running the program by using the Run dialog box or Windows Explorer.

Deploying Applications Through Group Policy

You can make applications available to users over the network through Group Policy. When you use Group Policy to deploy applications, you have two distribution options:

- The first option is to assign the application to users or computers. When an application is assigned to a computer, it is installed the next time the

computer is started and is available to all users of that computer the next time users log on. When an application is assigned to a user, it is installed the next time the user logs on to the network. An assigned application can also be configured to be installed on first use. In this configuration, the application is made available through shortcuts on the user's desktop or Start menu. With install-on-first-use configured, the application is installed when the user clicks a shortcut to launch the application.

- The second option is to publish the application and make it available for installation. When you publish an application, the application can be made available through extension activation. With extension activation configured, the program is installed when a user opens any file with an extension associated with the application. For example, if a user double-clicks a file with a .doc or .docx extension, Microsoft Word could be installed automatically.

You deploy applications for computers using a Microsoft Windows Installer Package (.msi file) and policies under Computer Configuration\Policies\Software Settings\Software Installation. You deploy applications for users using a Windows Installer Package (.msi file) and policies under User Configuration\Policies\Software Settings\Software Installation. The basic steps required to deploy applications through Group Policy are as follows:

1. For clients to access the Windows Installer Package, it must be located on a network share. As necessary, copy the Windows Installer Package (.msi file) to a network share that is accessible by the appropriate users.
2. In the Group Policy Management Editor, open the Group Policy object (GPO) from which you want to deploy the application. After it is deployed, the application is available to all clients to which the GPO applies. This means the application is available to computers and users in the related domain, site, or organizational unit (OU).
3. Expand Computer Configuration\Policies\Software Settings or User Configuration\Policies\Software Settings, right-click Software Installation, point to New, and then click Package.
4. Use the Open dialog box to locate the Windows Installer Package (.msi file) for the application, and then click Open. You are then given the choice to select the deployment method: Published, Assigned, or Advanced.
5. To publish or assign the program, select Published or Assigned, and then click OK. If you are configuring computer policy, the program is available the next time a computer affected by the GPO is started. If you are configuring user policy, the program is available to users in the domain, site, or OU the next time users log on. Currently logged-on users need to log off and then log on.
6. To configure additional deployment options for the program, select Advanced. You can then set additional deployment options as necessary.

Configuring Program Compatibility

If you want to install 16-bit or MS-DOS-based programs, you might need to make special considerations. Additionally, to get older programs to run, you might sometimes need to adjust compatibility options. Techniques for handling these situations are discussed in the following sections.

Special Installation Considerations for 16-Bit and MS-DOS-Based Programs

Many 16-bit and MS-DOS-based programs that don't require direct access to hardware can be installed and run on Windows 7 without any problems. However, most 16-bit and MS-DOS-based programs do not support long file names. To help ensure compatibility with these programs, Windows 7 maps long and short file names as necessary. This ensures that long file names are protected when they are modified by a 16-bit or an MS-DOS-based program. Additionally, it is important to note that some 16-bit and MS-DOS-based programs require 16-bit drivers, which are not supported on Windows 7. As a result, these programs won't run.

Most existing 16-bit and MS-DOS-based programs were originally written for Windows 3.0 or Windows 3.1. Windows 7 runs these older programs using a virtual machine that mimics the 386-enhanced mode used by Windows 3.0 and Windows 3.1. Unlike on other recent releases of Windows, on Windows 7 each 16-bit and MS-DOS-based application runs as a thread within a single virtual machine. This means that if you run multiple 16-bit and MS-DOS-based applications, they all share a common memory space. Unfortunately, if one of these applications hangs or crashes, it usually means the others will as well.

You can help prevent one 16-bit or MS-DOS-based application from causing others to hang or crash by running it in a separate memory space. To do this, follow these steps.

1. Right-click the program's shortcut icon, and then click Properties. If the program doesn't have a shortcut, create one, and then open the shortcut's Properties dialog box.
2. On the Shortcut tab, click the Advanced button. This displays the Advanced Properties dialog box.
3. Select the Run In Separate Memory Space check box.
4. Click OK twice to close all open dialog boxes and save the changes.

NOTE Running a program in a separate memory space uses additional memory. However, you'll usually find that the program is more responsive. Another added benefit is that you are able to run multiple instances of the program—as long as all the instances are running in separate memory spaces.

TIP The Windows 7 command prompt (Cmd.exe) is a 32-bit command prompt. If you want to invoke a 16-bit MS-DOS command prompt, you can use Command.com. Type **command** in the Run dialog box.

Forcing Program Compatibility

Some programs won't install or run on Windows 7 even if they work on previous versions of the Windows operating system. If you try to install a program that has known compatibility problems, Windows 7 should display a warning prompt telling you about the compatibility issue. In most cases, you should not continue installing or running a program with known compatibility problems, especially if the program is a system utility such as an antivirus program or a disk partitioning program, because running an incompatible system utility can cause serious problems. Running other types of incompatible programs can also cause problems, especially if they write to system locations on disk.

That said, if a program will not install or run on Windows 7, you might be able to run the program by adjusting its compatibility settings. Windows 7 provides two mechanisms for managing compatibility settings. You can use the Program Compatibility wizard, or you can edit the program's compatibility settings directly by using the program's Properties dialog box. Both techniques work the same way. However, the Program Compatibility wizard is the only way you can change compatibility settings for programs that are on shared network drives, CD or DVD drives, or other types of removable media drives. As a result, you can sometimes use the Program Compatibility wizard to install and run programs that would not otherwise install and run.

Using the Program Compatibility Wizard

You can only configure compatibility settings for programs you've installed. You can't configure compatibility settings for programs included with the operating system. To try to automatically detect compatibility issues using the Program Compatibility wizard, follow these steps.

1. Locate the program shortcut by navigating the menus under Start, All Programs. Right-click the program shortcut, and then click Troubleshoot Compatibility. This starts the Program Compatibility wizard, shown in Figure 9-2.

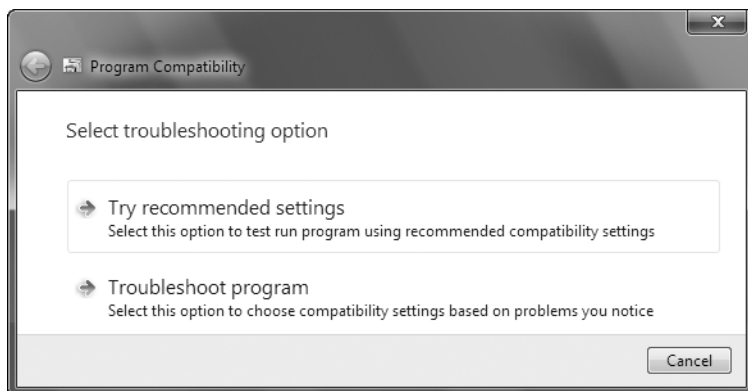


FIGURE 9-2 Troubleshoot program compatibility issues.

2. The wizard automatically tries to detect compatibility issues. To try to run the program you are troubleshooting with the recommended fixes, click Try Recommended Settings. Next, review the settings that will be applied, and then click Start The Program.
3. After running the program, click Next, and then do one of the following:
 - Click Yes, Save These Settings For This Program if the compatibility settings resolved the problem and you want to keep the settings.
 - Click No, Try Again Using Different Settings if the compatibility settings didn't resolve the problem and you want to repeat this process from the beginning.
 - Click No, Report The Problem To Microsoft And Check Online For A Solution if the compatibility settings didn't resolve the problem and you'd like to check for an online solution.
 - Click Cancel if you want to discard the compatibility settings and exit the wizard.

To perform advanced troubleshooting and use the Program Compatibility wizard to specify the compatibility settings to use, follow these steps:

1. Locate the program shortcut by navigating the menus under Start, All Programs. Right-click the program shortcut, and then click Troubleshoot Compatibility. This starts the Program Compatibility wizard.
2. Click Troubleshoot Program. On the What Problems Do You Notice? page, you can specify information about problems you've seen. The selections you make determine the wizard pages you see when you click Next. They include the following:
 - **The Program Worked On Earlier Versions Of Windows But Won't Install Or Run Now** If you select this option, you are prompted on one of the subsequent wizard pages to specify which version. Because your

choice sets the compatibility mode, choose the operating system for which the program was designed. When running the program, Windows 7 simulates the environment for the specified operating system.

- **The Program Opens But Doesn't Display Correctly** If you are trying to run a game, an educational program, or any other program that requires specific display settings, such as a program designed for Windows 98, you can select this option and then choose the type of display problem you are seeing. Your selections restrict the video display: when you use 256 colors, 640 × 480 screen resolution, or both, Windows restricts the video display. This can help with programs that have problems running at higher screen resolutions and greater color depths. Your selections can also disable themes, desktop compositing (which prevents special visual effects on the desktop), and display scaling of high dots-per-inch (DPI) settings.
 - **The Program Requires Additional Permissions** If you choose this option, the program will be configured to run with administrator privileges.
 - **I Don't See My Problem Listed** If you choose this option, the wizard displays optional pages for operating system and display issue selection. The wizard also sets the program to run as an administrator. Ultimately, choosing this option has the same effect as if you had selected all three of the previous options.
3. Review the compatibility settings that will be applied. If you don't want to apply these settings, click Cancel and repeat this procedure to select different options. If you want to apply these settings, click Start The Program, and the wizard runs the program with the compatibility settings you specified.
 4. After running the program, click Next to continue. When you continue, you are prompted to confirm whether the changes fixed the problem. Do one of the following:
 - If the compatibility settings resolved the problem and you want to keep the settings, click Yes, Save These Settings For This Program.
 - If the compatibility settings didn't resolve the problem and you want to repeat this process from the beginning, click No, Try Again Using Different Settings.
 - If the compatibility settings didn't resolve the problem and you'd like to check for an online solution, click No, Report The Problem To Microsoft And Check Online For A Solution.
 - If you want to discard the compatibility settings and exit the wizard, click Cancel.

NOTE If you've configured alternate display settings for an application, the application will run in the alternate display mode whenever you start it. To restore the original display settings, simply exit the program.

Setting Compatibility Options Directly

If a program you have already installed won't run correctly, you might want to edit the compatibility settings directly rather than through the wizard. To do this, follow these steps.

1. Right-click the program's shortcut icon, and then click Properties.
2. In the Properties dialog box, click the Compatibility tab. Any option you select is applied to the currently logged-on user for the application shortcut. To apply the setting to all users on the computer and regardless of which shortcut is used to start the application, click Change Setting For All Users to display the Properties dialog box for the application's .exe file, and then select the compatibility settings that you want to use for all users who log on to the computer.

NOTE Programs that are part of the Windows 7 operating system cannot be run in Compatibility mode. The options on the Compatibility tab are not available for built-in programs.

3. Select the Run This Program In Compatibility Mode For check box, and then use the selection menu to choose the operating system for which the program was designed.
4. If necessary, use the options in the Settings panel to restrict the video display settings for the program. Select 256 colors, 640 × 480 screen resolution, or both, as required.
5. If necessary, you can also disable visual themes, desktop compositing, and display scaling of high DPI settings.
6. Click OK. Double-click the shortcut to run the program and test the compatibility settings. If you still have problems running the program, you might need to modify the compatibility settings again.

Managing Installed and Running Programs

Windows 7 provides several management tools for working with programs. These tools include:

- **Task Manager** Provides options for viewing and managing running programs as well as options for viewing resource usage and performance
- **Programs** Provides tasks for viewing installed programs, adding and removing programs, viewing installed updates, and more
- **Default Programs** Helps you track and configure global default programs for the computer, personal default programs for individual users, AutoPlay settings for multimedia, and file associations for programs

- **Windows Features** Helps you view and manage the Windows components installed on a computer
- **Assoc** Helps you view and manage file type associations
- **Ftype** Helps you view and manage file type definitions

These tools and related configuration options are discussed in the sections that follow.

Managing Currently Running Programs

In Windows 7, you can view and work with a computer's currently running programs and processes by using Task Manager. You can open Task Manager by pressing Ctrl+Alt+Delete and then selecting Start Task Manager. As Figure 9-3 shows, Task Manager has two tabs for working with running programs:

- **Applications** Lists applications that are currently running in the foreground by name and status (such as Running or Not Responding). To exit a program, which might be necessary when it is not responding, click the program in the Task list, and then click End Task.
- **Processes** Lists all background and foreground applications running on the computer by image name, user name, and resource usage. To stop a process, click the process, and then click End Process.

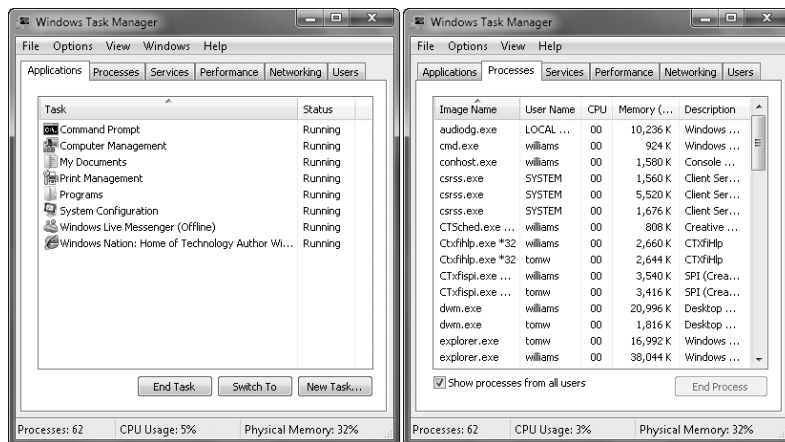


FIGURE 9-3 Use Task Manager to work with running applications and processes.

While the details for process count, CPU usage, and physical memory usage are for the computer as a whole, the processes are only listed for the currently logged-on user and the operating system by default. To see running processes for all users, you must click Show Processes From All Users.

TIP On the Processes tab, you can manage processes in additional ways by right-clicking a process and selecting from an extended list of options. The options include Open File Location, which opens the folder containing the executable file for the process in Windows Explorer; End Process Tree, which stops the process and all dependent processes; Create Dump File, which creates a memory dump file for the selected process; and Properties, which opens the Properties dialog box for the executable file.

Managing, Repairing, and Uninstalling Programs

Windows 7 considers any program you've installed on a computer or made available for a network installation to be an installed program. In Windows XP and earlier versions, you use the Add Or Remove Programs utility to install and manage applications. In Windows 7, you use the setup program that comes with the application to install applications, and you use the Installed Programs page in Control Panel to manage applications.

You can use the Installed Programs page to view, add, remove, or repair installed programs by following these steps:

1. Click Start, and then click Control Panel. In Control Panel, click Programs.
2. Click Programs And Features. You should see a list of installed programs.
3. In the Name list, right-click the program you want to work with, and then click one of the following commands:
 - **Uninstall** to uninstall the program
 - **Change** to modify the program's configuration
 - **Repair** to repair the program's installation

When you are uninstalling programs, keep the following in mind:

- Windows warns you if you try to uninstall a program while other users are logged on. Generally, you should be sure that other users are logged off before uninstalling programs. Otherwise, you might cause other users to lose data or experience other problems.
- Windows will allow you to remove only those programs that were installed with a Windows-compatible setup program. Although most applications have a setup program that uses InstallShield, Wise Install, or Microsoft Windows Installer, older programs might have a separate uninstall utility. Some older programs work by copying their data files to a program folder. In this case, you uninstall the program by deleting the related folder.
- Many uninstall programs leave behind data either inadvertently or by design. As a result, you often find folders for these applications within the Program Files folder. You could delete these folders, but they might contain important data files or custom user settings that could be used again if you reinstall the program.

- Sometimes, the uninstall process fails. Often, you can resolve any problem simply by rerunning the uninstaller for the program. Occasionally, you might need to clean up after the uninstall process. This might require removing program files and deleting remnants of the program in the Windows registry. A program called the Windows Installer Cleanup utility can help you clean up the registry. For more information on the utility and to download the software, see the article on the Microsoft support Website at <http://support.microsoft.com/kb/290301>.

Designating Default Programs

Default programs determine which programs are used with which types of files and how Windows handles files on CDs, DVDs, and portable devices. You configure default programs based on the types of files those programs support, either globally for all users of a computer or only for the current user. Individual user defaults override global defaults. For example, you could select Windows Media Player as the global default for all types of files it supports, and then all users of the computer would use Windows Media Player to play the sound, audio, and video files it supports. If a specific user wanted to use Apple iTunes instead as the default player for sound and audio files, you could configure iTunes to be that user's default player for the types of media files it supports.

You can configure global default programs for all the users of a computer by following these steps:

1. Click Start, and then click Control Panel. In Control Panel, click Programs.
2. Click Default Programs, and then click Set Program Access And Computer Defaults. You'll see the dialog box shown in Figure 9-4.

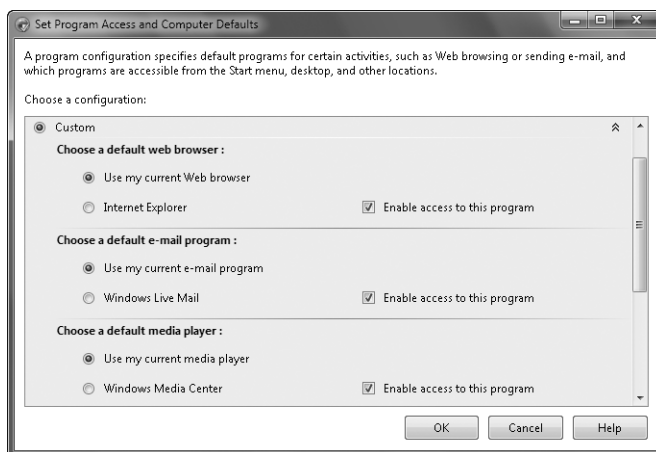


FIGURE 9-4 Choose a global default configuration.

3. Choose a configuration from one of the following options:

- **Microsoft Windows** Sets the currently installed Windows programs as the default programs for browsing the Web, sending e-mail, playing media files, and so on.
- **Non-Microsoft** Sets the currently installed programs as the default programs for browsing the Web, sending e-mail, playing media files, and so on.
- **Custom** Enables you to choose programs as the defaults for browsing the Web, sending e-mail, playing media files, and so on.

4. Click OK to save the settings.

To override global defaults, you can set default programs for individual users. You can configure default programs for the current user by following these steps:

1. Click Start, and then click Control Panel. In Control Panel, click Programs.
2. Click Default Programs, and then click Set Your Default Programs.
3. Select a program you want to work with in the Programs list.
4. If you want the program to be the default for all the file types and protocols it supports, click Set This Program As Default.
5. If you want the program to be the default for specific file types and protocols, click Choose Defaults For This Program. Select the file extensions for which the program should be the default, and then click Save.

Managing the Command Path

Windows uses the command path to locate executables. You can view the current command path for executables by using the PATH command. In a command shell, type **path** on a line by itself, and then press Enter. In a Windows PowerShell console, type **\$env:path** on a line by itself, and then press Enter. In the output, observe that Windows uses a semicolon (;) to separate individual paths, marking where one file path ends and another begins.

The command path is set during logon by using system and user environment variables. The path defined in the PATH system variable sets the base path. The path defined in the PATH user variable adds to the base path by using the following syntax:

`%PATH%;AdditionalPaths`

Here, %PATH% tells Windows to insert the current system paths, and *AdditionalPaths* designates the additional user-specific paths to use.

CAUTION An improperly set path can cause severe problems. You should always test any command path change before using it in a live environment. The command path is set during logon. Therefore, you must log off and then log on again to see the effects of the revised path.

Don't forget about the search order that Windows uses. Paths are searched in order, with the last path in the PATH user variable being the last one searched. This can sometimes slow the execution of your programs and scripts. To help Windows find your programs and scripts faster, you should consider placing a required path earlier in the search order.

Be careful when setting the command path. It is easy to overwrite all path information accidentally. For example, if you don't specify %PATH% when setting the user path, you will delete all other path information. One way to ensure that you can easily re-create the command path is to keep a copy of the command path in a file.

- When you are working with the command prompt, you can write the current command path to a file by entering **path > orig_path.txt**. Keep in mind that if you are using a standard command prompt rather than an administrator command prompt, you won't be able to write to secure system locations. In this case, you can write to a subdirectory to which you have access or to your personal profile. To write the command path to the command-shell window, type **path**.
- When you are working with the PowerShell console, you can write the current command path to a file by entering **\$env:path > orig_path.txt**. If you are using a standard console rather than an administrator console, you won't be able to write to secure system locations. In this case, you can write to a subdirectory to which you have access or to your personal profile. To write the command path to the PowerShell window, type **\$env:path**.

At the command prompt or in the PowerShell window, you can modify the command path by using the Setx.exe utility. You also can edit the command path by completing the following steps:

1. In Control Panel, click System And Security, and then click System.
2. In the System console, click Change Settings, or click Advanced System Settings in the left pane.
3. On the Advanced tab in the System Properties dialog box, click the Environment Variables button.
4. Select the PATH variable in the System Variables list. Under System Variables, click Edit.
5. By default, the path value is selected. Without pressing any other key, press the Right Arrow key. This should remove the selection highlight and place the insertion point at the end of the variable value.
6. Type a semicolon, and then enter a path to insert. Repeat as necessary, and then click OK three times.

In Group Policy, you can use a preference item to modify the command path. Follow these steps:

1. Open a Group Policy object (GPO) for editing in the Group Policy Management Editor. To configure preferences for computers, expand Computer

Configuration\Preferences\Windows Settings, and then select Environment. To configure preferences for users, expand User Configuration\Preferences\Windows Settings, and then select Environment.

2. Right-click the Environment node, point to New, and then click Environment Variable. This opens the New Environment Properties dialog box.
3. In the Action list, select Update to update the path variable, or select Replace to delete and then re-create the path variable. Next, select User Variable to work with user variables.
4. In the Name field, type **Path**. In the Value field, type the variable value. Typically, you'll enter **%PATH%**; followed by the paths you want to add, using a semicolon to separate each path. If the affected computers have existing PATH user variable definitions, you must provide the related paths to ensure that these paths are retained.
5. Use the options on the Common tab to control how the preference is applied. In most cases, you'll want to create the PATH variable only once (rather than have Group Policy re-create the variable each time policy is refreshed). If so, select Apply Once And Do Not Reapply.
6. Click OK. The next time policy is refreshed, the preference item will be applied as appropriate for the GPO in which you defined the preference item.

CAUTION Incorrectly setting the path can cause serious problems. Before deploying an updated path to multiple computers, you should test the configuration. One way to do this is to create a GPO in Active Directory that applies only to an isolated test computer. Next, create a preference item for this GPO, and then wait for a policy to refresh or apply policy using GPOUpdate. If you are logged on to the computer, you need to log off and then log back on before you can confirm the results.

Managing File Extensions and File Associations

File extensions and file associations also are important for determining how programs run. The types of files that Windows considers to be executables are determined by the file extensions for executables. File extensions allow users to execute a command by using just the command name. File associations are what allow users to double-click a file and open the file automatically in a related application. Two types of file extensions are used:

- **File extensions for executables** Executable files are defined with the %PATHEXT% environment variable and can be set using the Environment Variables dialog box or with Group Policy preference items in much the same way as the PATH variable. You can view the current settings by typing **set pathext** at the command line or by typing **\$env:pathext** at a PowerShell prompt. The default setting is PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC. With this setting, the command line knows which files

are executable and which files are not, so you don't have to specify the file extension at the command line.

- **File extensions for applications** File extensions for applications are referred to as file associations. File associations are what enable you to pass arguments to executables and to open documents, worksheets, or other application files by double-clicking their file icons. Each known extension on a system has a file association that you can view at a command prompt by typing **assoc** followed by the extension, such as **assoc .doc** or **assoc .docx**. Each file association in turn specifies the file type for the file extension. This can be viewed at a command prompt by typing **ftype** followed by the file association, such as **ftype Word.Document.8** or **ftype Word.Document.12**.

NOTE **Assoc** and **Ftype** are internal commands for the command shell (Cmd.exe). To use the **Assoc** command in PowerShell, enter **cmd /c assoc** followed by the extension, such as **cmd /c assoc .doc**. To use the **Ftype** command in PowerShell, enter **cmd /c ftype** followed by the file association, such as **cmd /c ftype Word.Document.8**.

With executables, the order of file extensions in the %PATHEXT% variable sets the search order used by the command line on a per-directory basis. Thus, if a particular directory in the command path has multiple executables that match the command name provided, a .com file would be executed before an .exe file and so on.

Every known file extension on a system has a corresponding file association and file type—even extensions for executables. In some cases, the file type is the extension text without the period followed by the keyword file, such as cmdfile, exefile, or batfile, and the file association specifies that the first parameter passed is the command name and that other parameters should be passed on to the application. For example, if you type **assoc .exe** to see the file associations for .exe executables, you then type **ftype exefile**. You'll see the file association is set to the following:

```
exefile="%1" %*
```

Thus, when you run an .exe file, Windows knows the first value is the command that you want to run and anything else provided is a parameter to pass along.

File associations and types are maintained in the Windows registry and can be set using the **Assoc** and **Ftype** commands, respectively. To create the file association at the command line, type **assoc** followed by the extension setting, such as **assoc .pl=perfile**. To create the file type at the command line, set the file-type mapping, including how to use parameters supplied with the command name, such as **ftype perfile=C:\Perl\Bin\Perl.exe "%1" %***.

You also can associate a file type or protocol with a specific program by completing the following steps:

1. Click Start, and then click Control Panel. In Control Panel, click Programs.
2. Click Default Programs, and then click Associate A File Type Or Protocol With A Program.

3. On the Set Associations page, current file associations are listed by file extension and the current default for that extension. To change the file association for an extension, click the file extension, and then click Change Program.
4. Do one of the following:
 - The Recommended Programs list shows programs that are registered in the operating system as supporting files with the selected extension. Click a recommended program to set it as the default for the selected extension, and then click OK.
 - The Other Programs list shows programs that might also support the selected extension. Click a program to set it as the default for the selected extension, and then click OK. Alternatively, click Browse to locate another program to use as the default.

In Group Policy, you can use a preference item to create new file types and file associations. To create a preference item for a new file type, follow these steps:

1. Open a Group Policy object (GPO) for editing in the Group Policy Management Editor. Expand Computer Configuration\Preferences\Control Panel Settings, and then select Folder Options.
2. Right-click the Folder Options node, point to New, and then click File Type. This opens the New File Type Properties dialog box.
3. In the Action list, select Create, Update, Replace, or Delete.
4. In the File Name Extension field, type the extension of the file type without the period, such as **pl**.
5. In the Associated Class list, select a registered class to associate with the file type.
6. Use the options on the Common tab to control how the preference is applied. In most cases, you'll want to create the new variable only once. If so, select Apply Once And Do Not Reapply.
7. Click OK. The next time policy is refreshed, the preference item will be applied as appropriate for the GPO in which you defined the preference item.

To create a preference item for a new file association, follow these steps:

1. Open a Group Policy object (GPO) for editing in the Group Policy Management Editor. Expand User Configuration\Preferences\Control Panel Settings, and then select Folder Options.
2. Right-click the Folder Options node, point to New, and then click Open With. This opens the New Open With Properties dialog box.
3. In the Action list, select Create, Update, Replace, or Delete.
4. In the File Name Extension field, type the extension of the file type without the period, such as **pl**.

5. Click the Browse (...) button to the right of the Associated Program field, and then use the Open dialog box to select the program to associate with the file type.
6. Optionally, select Set As Default to make the associated program the default for files with the previously specified file extension.
7. Use the options on the Common tab to control how the preference is applied. In most cases, you'll want to create the new variable only once. If so, select Apply Once And Do Not Reapply.
8. Click OK. The next time policy is refreshed, the preference item will be applied as appropriate for the GPO in which you defined the preference item.

Configuring AutoPlay Options

In Windows 7, AutoPlay options determine how Windows handles files on CDs, DVDs, and portable devices. You can configure separate AutoPlay options for each type of CD, DVD, and media your computer can handle by following these steps:

1. Click Start, and then click Control Panel. In Control Panel, click Programs.
2. Click Default Programs, and then click Change AutoPlay Settings. This displays the AutoPlay page in Control Panel.
3. As shown in Figure 9-5, use the media selection list to set the default AutoPlay option for each media type.

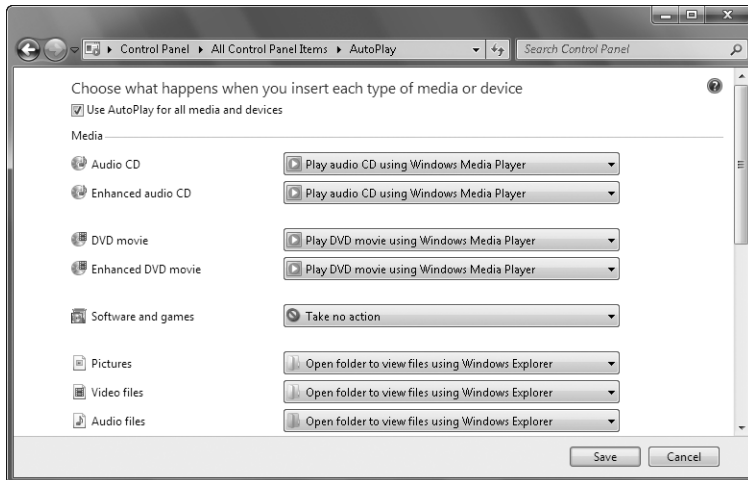


FIGURE 9-5 Set AutoPlay options for CDs, DVDs, and portable devices.

4. Click Save to save your settings.

Adding and Removing Windows Features

In Windows XP and earlier versions of Windows, you use the Add/Remove Windows Components option of the Add Or Remove Programs utility to add or remove operating system components. In Windows Vista and Windows 7, operating system components are considered Windows features that can be turned on or off rather than added or removed.

You can turn on or off Windows features by following these steps:

1. Click Start, and then click Control Panel. In Control Panel, click Programs.
2. Under Programs And Features, click Turn Windows Features On Or Off. This displays the Windows Features dialog box.
3. As shown in Figure 9-6, select the check boxes for features to turn them on, or clear the check boxes for features to turn them off.

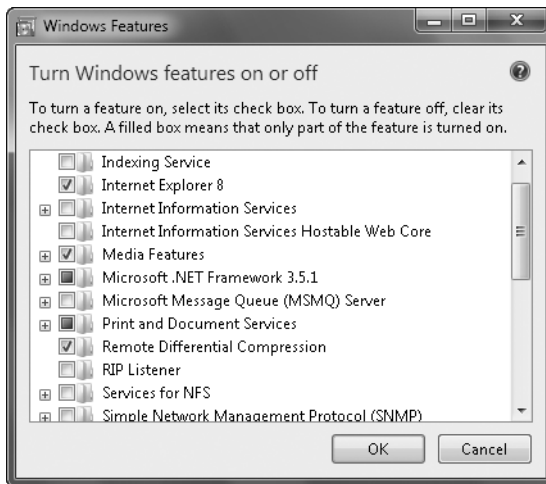


FIGURE 9-6 Add or remove operating system components.

4. Click OK, and Windows 7 reconfigures components for any changes you made.