AutoPlay

This section provides information about the AutoPlay feature. Information is divided into the following groups:

- About AutoPlay
- AutoPlay Overview
- AutoPlay Reference

About AutoPlay

Microsoft® AutoPlay is a feature of the Microsoft Windows® operating system. AutoPlay automates the procedures for installing and configuring products designed for Windows-based platforms that are distributed on compact discs. When you insert a disc containing AutoPlay into a CD-ROM drive on a computer running Windows, AutoPlay automatically starts an application on the disc that installs, configures, and runs the selected product.

You can use AutoPlay to install and run CD-ROM applications that run in Windows, whether the application is based on the MS-DOS® operating system, Windows 3.0, Windows 3.1, Windows 95, or Windows NT®. If you want your CD-ROM product to display the Microsoft Windows 95 logo, it must be enabled for AutoPlay.

Note

MS-DOS, Windows versions prior to Windows 95, and Windows NT versions 3.51 and earlier do not support AutoPlay. Adding AutoPlay to a compact disc, however, does not hinder or alter user interaction on computers running one of these operating systems.

AutoPlay Overview

This topic contains general information about the AutoPlay component. The following topics are discussed:

- How AutoPlay Works
- Autorun.inf
- Tips for Writing AutoPlay Applications
- Suppressing AutoPlay
- AutoPlay for MS-DOS-Based Applications

How AutoPlay Works

The implementation of AutoPlay relies on the following items:

- *A set of 32-bit CD-ROM device drivers for Windows 95 and Windows NT*. These device drivers detect when a user inserts a compact disc into a CD-ROM drive; device drivers for MS-DOS or previous versions of Windows do not.
- *An Autorun.inf file on the compact disc.* When you insert a disc into a CD-ROM drive on a computer running Windows 95 or Windows NT, the system immediately checks to see if the disc has a personal computer file system. If it does, the system searches for a file named Autorun.inf. This file specifies the application that AutoPlay runs. It can contain other information as well. or more information, see Autorun.inf.
- *A startup application on the compact disc.* Although you can start any application on the disc by specifying it in the Autorun.inf file, typically the application performs a startup or installation function. By including your own startup application, you can control the install, uninstall, and run processes for your product.

Autorun.inf

The Autorun.inf file is a text file located in the root directory of the CD on which your DirectX® application is shipped. This file contains the name of the startup program on the disc. This startup program runs automatically when the disc is inserted in the CD-ROM drive. The Autorun.inf file also contains the name of the file of the icon that you want to represent your application's CD in the Windows user interface. In addition, the Autorun.inf file can contain optional menu commands that you want added to the shortcut menu. These menu commands are displayed when the user right-clicks the CD-ROM icon.

The following is an example of a minimal Autorun.inf file.

```
[autorun]
open=filename.exe
icon=filename.ico
```

The **[autorun]** section identifies the lines that follow it as AutoPlay commands. An **[autorun]** section is required in every Autorun.inf file. The **open** command specifies the path and file name of the startup application. The **icon** command specifies the file name that contains the icon.

The Autorun.inf file also can contain architecture-specific sections for Windows NT 4.0 running on RISC processors. For each type of processor architecture, add a section to the Autorun.inf file that contains the file name of the startup application you want to run for that architecture. The following table lists the commands used for the architectures that AutoPlay supports.

Architecture	Section Title
368 or higher	[autorun]
MIPS	[autorun.mips]
DEC Alpha	[autorun.alpha]
PowerPC	[autorun.ppc]

The following example shows how to create an Autorun.inf file that runs different startup applications depending on the computer architecture:

```
[autorun]
open=filename.exe
icon=filename.ico

[autorun.mips]
open=filenam2.exe
icon=filename.ico

[autorun.alpha]
open=filenam3.exe
icon=filename.ico

[autorun.ppc]
open=filenam4.exe
icon=filename.ico
```

The shell checks for an architecture-specific section first. If it does not find one, it uses the information in the **[autorun]** section. After the shell finds a section, it ignores all the other sections, so each section must contain all the information for that architecture.

Tips for Writing AutoPlay Applications

This section presents helpful guidelines for writing AutoPlay applications:

- Opening a Startup Application
- Loading in the Background
- Conserving Hard Disk Space
- Using the Registry
- Setting the NoDriveTypeAutoRun Value

Opening a Startup Application

Users should receive feedback soon after they insert an AutoPlay compact disc into the disc drive. Therefore, your startup application should be a small program that loads quickly. The startup application should clearly identify the title it plays and provide an easy way to cancel the operation.

Loading in the Background

Typically, the startup application presents users with a dialog box asking them if they would like to proceed. The user normally clicks an **OK** button to continue. Take advantage of the time the user spends reading the dialog box by starting another thread that begins loading the setup application. If the user clicks **OK**, your setup program will already be loading. This significantly reduces the user's perception of the amount of time it takes to load your application.

Conserving Hard Disk Space

Hard disk space is a limited resource. Here are a few hints for minimizing hard disk usage:

- *Don't put files on the user's hard drive.* Run your application from the compact disc directly, without running any installation application.
- *Keep installed files to a minimum.* If your application needs to use the hard disk, install only the functional components necessary to run the application. In addition, provide a way to uninstall these components from the hard disk. For more information about uninstalling an application, see the documentation included with the Microsoft Platform Software Development Kit (SDK).
- *Provide the user with caching control.* If your application needs to use the drive as a data cache, provide the user with options in the startup application that will discard the cached data when the user quits the title or game.

Using the Registry

The registry is a feature of Windows that supersedes the initialization (.ini) and application configuration files. For information about application programming interfaces that manipulate the registry, see the documentation included with the Platform SDK.

If your product records and uses initialization information, you can use the registry to store and retrieve this information. Your startup application can use the information in the registry to determine whether the product needs to be installed. If there are no registry entries for your product—which means your product is being used for the first time—you could display a dialog box that lists the setup options. If your product is listed in the registry—which means it has already been installed—you could skip the setup options.

By changing the system registry, you can cause a computer to read the Autorun.inf file from a floppy disk. This feature of implementing AutoPlay on a floppy disk is

provided only to help you debug your Autorun.inf files before you burn the compact disc. AutoPlay is intended for public distribution on compact disc only. To implement AutoPlay on a floppy disk, carry out the following procedure:

1. In Registry Editor (Regedit.exe), click **Edit**, and then click **Find**.
2. In the **Find What** box, type the following, and then click **Find Next**:

   NoDriveTypeAutoRun
3. Click **Edit**, and then click **Modify**.
4. Change the data of the NoDriveTypeAutoRun value from 0000 95 00 00 00 to 0000 91 00 00 00, and then click **OK**.

   This enables AutoPlay on any drive. You must, however, start AutoPlay manually when it is installed on a floppy disk. To do this, double-click the floppy disk icon, or right-click the floppy disk icon, and then click **AutoPlay**.
5. After you complete your tests of Autorun.inf, reset the value of NoDriveTypeAutoRun to 0000 95 00 00 00.

Important

Because implementing AutoPlay on a floppy disk provides an easy way to spread computer viruses, it is appropriate to suspect that any publicly distributed floppy disk that contains Autorun.inf files is contaminated.

For more information about the NoDriveTypeAutoRun value, see Setting the NoDriveTypeAutoRun Value.

Setting the NoDriveTypeAutoRun Value

The NoDriveTypeAutoRun value in the registry is a 4-byte binary data value of the type **REG_BINARY**. The first byte of this value represents different kinds of drives that can be excluded from working with AutoPlay. The initial setting for this byte is 0x95, which excludes the unrecognized type drive, DRIVE_UNKNOWN, DRIVE_REMOVEABLE, and DRIVE_FIXED media types from being used with AutoPlay. You can enable a floppy disk drive for AutoPlay by resetting bit 2 to zero, or by specifying the value 0x91 to maintain the rest of the initial settings. For information about how to change the registry values, see Using the Registry. A table identifying the bits, bitmask constants, and a brief description of the drives follows:

Bit number	Bitmask constant	Description
0 (low-order bit)	DRIVE_UNKNOWN	Drive type not identified.
1	DRIVE_NO_ROOT_DIR	Root directory does not exist.
2	DRIVE_REMOVEABLE	Disk can be removed from drive (a floppy disk).
3	DRIVE_FIXED	Disk cannot be removed from drive (a hard disk).
4	DRIVE_REMOTE	Network drive.

5	DRIVE_CDROM	CD-ROM drive.
6	DRIVE_RAMDISK	RAM disk.
7 (high-order bit)		Reserved for future use.

Note

For Windows NT, you must restart Windows NT Explorer before any changes take effect.

Suppressing AutoPlay

There are a variety of reasons why you might want to suppress AutoPlay. One might be that your application has a setup program that requires the user to insert a disc which contains an Autorun.inf file. In this case, you would not want the AutoPlay feature to begin running an application while your setup application is running.

Another reason to suppress AutoPlay might be that your user needs to do disk swapping during your game. You would probably not want the setup program to execute while your game is in progress.

You can manually prevent the Autorun.inf file on a compact disc from being parsed and carried out by holding down the SHIFT key when you insert the disc.

Users of Windows NT version 4.0 can suppress AutoPlay automatically using the code example shown below. This initialization code is based on the assumption that your setup application is in the foreground window.

```
uMessage - RegisterWindowMessage(TEXT("QueryCancelAutoPlay"));
```

Then, add the following code to your setup window procedure:

```
if(msg == uMessage)
{
   // return 1 to cancel AutoPlay
   // return 0 to allow AutoPlay
   return 1L;
}
```

Windows 95 users can suppress AutoPlay by setting the value of the NoDriveTypeAutoRun registry entry to 0xFF. The code sample below demonstrates how this can be done. Insert this code near the beginning of the **WinMain** function.

```
const unsigned long WINDOWS_DEFAULT_AUTOPLAY_VALUE=0x095;
const unsigned long AUTOPLAYOFF=0x0FF;
   unsigned long ulOldAutoRunValue = WINDOWS_DEFAULT_AUTOPLAY_VALUE;
   unsigned long ulDisableAutoRun = AUTOPLAYOFF;
   unsigned long ulDataSize = sizeof(unsigned long);
   HKEY hkey = NULL;
char *lpzRegistryString = "Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer"
```

```
if( RegOpenKeyEx(HKEY_CURRENT_USER,
   lpzRegistryString ,
   0, KEY_ALL_ACCESS, &hkey ) == ERROR_SUCCESS )
{
   if( RegQueryValueEx(hkey,"NoDriveTypeAutoRun", 0, NULL,
                  (unsigned char*)&ulOldAutoRunValue,
                  &ulDataSize ) == ERROR_SUCCESS )
   {
      RegSetValueEx(hkey, "NoDriveTypeAutoRun", 0, REG_BINARY,
                  (const unsigned char*)&ulDisableAutoRun, 4);
   }
   else ulOldAutoRunValue = WINDOWS_DEFAULT_AUTOPLAY_VALUE;

   RegFlushKey( hkey );
   RegCloseKey( hkey );
}
```

Just before your application terminates, it must reset the registry to the old value, as shown in the code example below:

```
// Restore original AutoPlay settings.
if (RegOpenKeyEx(HKEY_CURRENT_USER,
               lpzRegistryString ,
               0, KEY_ALL_ACCESS, &hkey ) == ERROR_SUCCESS )
{
   RegSetValueEx(hkey, "NoDriveTypeAutoRun", 0, REG_BINARY,
               (const unsigned char*)&ulOldAutoRunValue,
                4 );
   RegFlushKey( hkey );
   RegCloseKey( hkey );
}
```

AutoPlay for MS-DOS-Based Applications

You also can use AutoPlay to install, configure, and run MS-DOS–based applications in a Windows MS-DOS session. You can even configure each MS-DOS–based application with its own unique icon, Config.sys file, and Autoexec.bat file.

Windows creates the correct configuration files for the MS-DOS–based application. The startup application then starts the MS-DOS–based application in a window.

AutoPlay Reference

This section contains reference information for the AutoPlay feature.

Commands

This section contains information about the following AutoPlay commands:

- **defaulticon**
- **icon**
- **open**
- **shell**
- **shell\verb**

defaulticon

The **defaulticon** command specifies an absolute path on the compact disc to the file that contains the information for the icon. The icon represents the AutoPlay-enabled CD in the Windows user interface.

```
defaulticon=path\iconname.ico
```

Parameters

path\iconname.ico
Absolute path and file name of the file containing the icon. The icon can be in a .ico, .bmp, .exe, or .dll file. If a file contains more than one icon, specify the resource number (index) of the icon in the file to use.

Remarks

If both the **icon** and **defaulticon** commands are present in an Autorun.inf file, AutoPlay uses the icon specified in the **defaulticon** command.

See Also

icon

icon

The **icon** command specifies a file that contains an icon which represents the AutoPlay-enabled CD in the Windows user interface. The file name specified with this command must be located in the same directory as the file name specified by the **open** command.

```
icon=filename.ico
```

Parameters

filename.ico

Name of the file containing the icon information. You also can specify a .bmp, .exe, or .dll file. If a file contains more than one icon, specify the resource number (index) of the icon in the file to use.

Remarks

The following example specifies the second icon in a file to represent a compact disc. The first icon's index is set to zero.

```
icon=filename.exe 1
```

See Also

defaulticon

open

The **open** command specifies the path and file name of the application that AutoPlay runs when you insert the compact disc in a CD-ROM drive.

```
open=dir\filename.exe
```

Parameters

dir\filename.exe

Path and file name of any executable file to run when the compact disc is inserted. If no path is specified, Windows looks for the file in the root directory on the compact disc. Specify a relative path to locate the file in a subdirectory.

Remarks

Use the **open** command to open a startup application that provides instant feedback to the user. For more information about startup applications, see Opening a Startup Application.

AutoPlay can pass command-line parameters to the application when it runs. Specify command line parameters after the filename.

shell

The **shell** command changes the default entry of the shortcut menu to the specified custom command.

```
shell=verb
```

Parameters

verb
: Abbreviated form of a custom command. The custom command must be defined in the Autorun.inf file.

Remarks

If the user right clicks the icon which represents your AutoPlay-enabled CD, a shortcut menu will be displayed. AutoPlay is the default menu item defined for any AutoPlay-enabled CD. The shell directive changes the default command to the specified command verb.

The command verb is executed when the user selects it from the shortcut menu. It is also executed when the user double-clicks the icon representing your CD.

See Also

shell\verb

shell\verb

The **shell\verb** command specifies a custom command listed in the shortcut menu for the icon. The first line identifies the executable file that performs the command. The second line specifies the custom entry of the shortcut menu.

```
shell\verb\command=filename.exe
shell\verb=Menu Item Name
```

Parameters

verb
: Abbreviated form of the command. This parameter associates a command with the executable file name and the menu item. It must not contain embedded spaces. You will not see *verb* on the shortcut menu unless *Menu Item Name* is omitted from the Autorun.inf file.

filename.exe
: File name of the application that performs the custom command.

Menu Item Name
: Menu item text that can contain mixed-case letters and spaces. You also can set a shortcut key for the menu item by preceding one of the letters in the item with an ampersand (&).

Remarks

When a user right-clicks an icon in the Windows user interface, a shortcut menu for that icon appears. If an Autorun.inf file is present on a CD and the user right-clicks the icon for the CD, Windows automatically adds AutoPlay to the shortcut menu for the disk's icon. It also sets AutoPlay as the default behavior. Double-clicking the icon starts whatever is specified in the **open** command.

To add the command **ReadMe** to the shortcut menu for your product and to make the letter "M" the shortcut key, include the following in the Autorun.inf file:

```
shell\readit\command=notepad abc\readme.txt
shell\readit=Read &Me
```

See Also

shell, **open**