Platform SDK: DLLs, Processes, and Threads

STARTUPINFO

The **STARTUPINFO** structure is used with the **CreateProcess**, **CreateProcessAsUser**, and **CreateProcessWithLogonW** functions to specify the window station, desktop, standard handles, and appearance of the main window for the new process.

```
typedef struct _STARTUPINFO {
  DWORD cb;
  LPTSTR lpReserved;
  LPTSTR lpDesktop;
  LPTSTR lpTitle;
  DWORD dwX;
  DWORD dwY;
  DWORD dwXSize;
  DWORD dwYSize;
  DWORD dwXCountChars;
  DWORD dwYCountChars;
  DWORD dwFillAttribute;
  DWORD dwFlags;
  WORD wShowWindow;
  WORD cbReserved2;
  LPBYTE lpReserved2;
  HANDLE hStdInput;
  HANDLE hStdOutput;
  HANDLE hStdError;
} STARTUPINFO, *LPSTARTUPINFO;
```

Members

cb

Size of the structure, in bytes.

lpReserved

Reserved. Set this member to NULL before passing the structure to **CreateProcess**.

lpDesktop

Pointer to a null-terminated string that specifies either the name of the desktop, or the name of both the desktop and window station for this process. A backslash in the string indicates that the string includes both the desktop and window station names.

For **CreateProcess** and **CreateProcessAsUser**, if this member is NULL, the new process inherits the desktop and window station of its parent process. If this member is an empty string, the process does not inherit the desktop and window station of its parent process; instead, the system determines if a new desktop and window station need to be created. If the impersonated user already has a desktop, the system uses the existing desktop.

For **CreateProcessWithLogonW**, if this member is NULL or an empty string, the new process inherits the desktop and window station of its parent process. **CreateProcessWithLogonW** adds permission for the specified user account to the inherited window station and desktop. Otherwise, if this member specifies a desktop, it is the responsibility of the application to add permission for the specified user account to the specified window station and desktop.

Windows Me/98/95: Desktops and window stations are not supported.

lpTitle

For console processes, this is the title displayed in the title bar if a new console window is created. If NULL, the name of the executable file is used as the window title instead. This parameter must be NULL for GUI or console processes that do not create a new console window.

dwX

If **dwFlags** specifies STARTF_USEPOSITION, this member is the x offset of the upper left corner of a window if a new window is created, in pixels. Otherwise, this member is ignored.

The offset is from the upper left corner of the screen. For GUI processes, the specified position is used the first time the new process calls **CreateWindow** to create an overlapped window if the *x* parameter of **CreateWindow** is CW_USEDEFAULT.

dwY

If **dwFlags** specifies STARTF_USEPOSITION, this member is the y offset of the upper left corner of a window if a new window is created, in pixels. Otherwise, this member is ignored.

The offset is from the upper left corner of the screen. For GUI processes, the specified position is used the first time the new process calls **CreateWindow** to create an overlapped window if the *y* parameter of **CreateWindow** is CW_USEDEFAULT.

dwXSize

If **dwFlags** specifies STARTF_USESIZE, this member is the width of the window if a new window is created, in pixels. Otherwise, this member is ignored.

For GUI processes, this is used only the first time the new process calls **CreateWindow** to create an overlapped window if the *nWidth* parameter of **CreateWindow** is CW_USEDEFAULT.

dwYSize

If **dwFlags** specifies STARTF_USESIZE, this member is the height of the window if a new window is created, in pixels. Otherwise, this member is ignored.

For GUI processes, this is used only the first time the new process calls **CreateWindow** to create an overlapped window if the *nHeight* parameter of **CreateWindow** is CW_USEDEFAULT.

dwXCountChars

If **dwFlags** specifies STARTF_USECOUNTCHARS, if a new console window is created in a console process, this member specifies the screen buffer width, in character columns. Otherwise, this member is ignored.

dwYCountChars

If **dwFlags** specifies STARTF_USECOUNTCHARS, if a new console window is created in a console process, this member specifies the screen buffer height, in character rows. Otherwise, this member is ignored.

dwFillAttribute

If **dwFlags** specifies STARTF_USEFILLATTRIBUTE, this member is the initial text and background colors if a new console window is created in a console application. Otherwise, this member is ignored.

This value can be any combination of the following values: FOREGROUND_BLUE, FOREGROUND_GREEN, FOREGROUND_RED, FOREGROUND_INTENSITY, BACKGROUND_BLUE, BACKGROUND_GREEN, BACKGROUND_RED, and BACKGROUND_INTENSITY. For example, the following combination of values produces red text on a white background:

```
FOREGROUND_RED| BACKGROUND_RED| BACKGROUND_GREEN| BACKGROUND_BLUE
```

dwFlags

Bit field that determines whether certain **STARTUPINFO** members are used when the process creates a window. This member can be one or more of the following values.

Value	**Meaning**
STARTF_FORCEONFEEDBACK	Indicates that the cursor is in feedback mode for two seconds after **CreateProcess** is called. The Working in Background cursor is displayed (see the Pointers tab in the Mouse control panel utility). If during those two seconds the process makes the first GUI call, the system gives five more seconds to the process. If during those five seconds the process shows a window, the system gives five more seconds to the process to finish drawing the window. The system turns the feedback cursor off after the first call to **GetMessage**, regardless of whether the process is drawing.
STARTF_FORCEOFFFEEDBACK	Indicates that the feedback cursor is forced off while the process is starting. The Normal Select cursor is displayed.
STARTF_RUNFULLSCREEN	Indicates that the process should be run in full-screen mode, rather than in windowed mode. This flag is only valid for console applications running on an x86 computer. **Windows Me/98/95:** This value is not supported.
STARTF_USECOUNTCHARS	If this value is not specified, the **dwXCountChars** and **dwYCountChars** members are ignored. **Windows Me/98/95:** This value is not supported.
STARTF_USEFILLATTRIBUTE	If this value is not specified, the **dwFillAttribute** member is ignored.
STARTF_USEPOSITION	If this value is not specified, the **dwX** and **dwY** members are ignored.
STARTF_USESHOWWINDOW	If this value is not specified, the **wShowWindow** member is ignored.
STARTF_USESIZE	If this value is not specified, the **dwXSize** and **dwYSize** members are ignored.
STARTF_USESTDHANDLES	Sets the standard input, standard output, and standard

error handles for the process to the handles specified in the **hStdInput**, **hStdOutput**, and **hStdError** members of the **STARTUPINFO** structure. For this to work properly, the handles must be inheritable and the **CreateProcess** function's *fInheritHandles* parameter must be set to TRUE. For more information, see Handle Inheritance.

If this value is not specified, the **hStdInput**, **hStdOutput**, and **hStdError** members of the **STARTUPINFO** structure are ignored.

wShowWindow

If **dwFlags** specifies STARTF_USESHOWWINDOW, this member can be any of the SW_ constants defined in Winuser.h. Otherwise, this member is ignored.

For GUI processes, **wShowWindow** specifies the default value the first time **ShowWindow** is called. The *nCmdShow* parameter of **ShowWindow** is ignored. In subsequent calls to **ShowWindow**, the **wShowWindow** member is used if the *nCmdShow* parameter of **ShowWindow** is set to SW_SHOWDEFAULT.

cbReserved2

Reserved for use by the C Runtime; must be zero.

lpReserved2

Reserved for use by the C Runtime; must be NULL.

hStdInput

If **dwFlags** specifies STARTF_USESTDHANDLES, this member is a handle to be used as the standard input handle for the process. Otherwise, this member is ignored.

hStdOutput

If **dwFlags** specifies STARTF_USESTDHANDLES, this member is a handle to be used as the standard output handle for the process. Otherwise, this member is ignored.

hStdError

If **dwFlags** specifies STARTF_USESTDHANDLES, this member is a handle to be used as the standard error handle for the process. Otherwise, this member is ignored.

Remarks

For graphical user interface (GUI) processes, this information affects the first window created by the **CreateWindow** function and shown by the **ShowWindow** function. For console processes, this information affects the console window if a new console is created for the process. A process can use the **GetStartupInfo** function to retrieve the **STARTUPINFO** structure specified when the process was created.

If a GUI process is being started and neither STARTF_FORCEONFEEDBACK or STARTF_FORCEOFFFEEDBACK is specified, the process feedback cursor is used. A GUI process is one whose subsystem is specified as "windows."

Requirements

Client: Included in Windows XP, Windows 2000 Professional, Windows NT Workstation, Windows Me, Windows 98, and Windows 95.
Server: Included in Windows Server 2003, Windows 2000 Server, and Windows NT Server.
Unicode: Declared as Unicode and ANSI structures.
Header: Declared in Winbase.h; include Windows.h.

See Also

Processes and Threads Overview, Process and Thread Structures, **CreateProcess**, **CreateProcessAsUser**, **CreateProcessWithLogonW**, **GetStartupInfo**

Platform SDK Release: February 2003

What did you think of this topic?   Order a Platform SDK CD