

# GetSystemMetrics function

 [docs.microsoft.com/en-us/windows/desktop/api/winuser/nf-winuser-getsystemmetrics](https://docs.microsoft.com/en-us/windows/desktop/api/winuser/nf-winuser-getsystemmetrics)

## In this article

- 08/17/2018
- 16 minutes to read

Retrieves the specified system metric or system configuration setting.

Note that all dimensions retrieved by **GetSystemMetrics** are in pixels.

## Syntax

```
int GetSystemMetrics(  
    int nIndex  
);
```

## Parameters

**nIndex**

Type: **int**

The system metric or configuration setting to be retrieved. This parameter can be one of the following values. Note that all SM\_CX\* values are widths and all SM\_CY\* values are heights. Also note that all settings designed to return Boolean data represent **TRUE** as any nonzero value, and **FALSE** as a zero value.

Value	Meaning
<b>SM_ARRANGE</b> 56	The flags that specify how the system arranged minimized windows. For more information, see the Remarks section in this topic.
<b>SM_CLEANBOOT</b> 67	The value that specifies how the system is started: <ul style="list-style-type: none"><li>• 0 Normal boot</li><li>• 1 Fail-safe boot</li><li>• 2 Fail-safe with network boot</li></ul> <p>A fail-safe boot (also called SafeBoot, Safe Mode, or Clean Boot) bypasses the user startup files.</p>
<b>SM_CMONITORS</b> 80	The number of display monitors on a desktop. For more information, see the Remarks section in this topic.
<b>SM_CMOUSEBUTTONS</b> 43	The number of buttons on a mouse, or zero if no mouse is installed.

<b>SM_CONVERTIBLESLATEMODE</b> 0x2003	Reflects the state of the laptop or slate mode, 0 for Slate Mode and non-zero otherwise. When this system metric changes, the system sends a broadcast message via <a href="#">WM_SETTINGCHANGE</a> with "ConvertibleSlateMode" in the LPARAM. Note that this system metric doesn't apply to desktop PCs. In that case, use <a href="#">GetAutoRotationState</a> .
<b>SM_CXBORDER</b> 5	The width of a window border, in pixels. This is equivalent to the SM_CXEDGE value for windows with the 3-D look.
<b>SM_CXCURSOR</b> 13	The width of a cursor, in pixels. The system cannot create cursors of other sizes.
<b>SM_CXDLGFRAME</b> 7	This value is the same as SM_CXFIXEDFRAME.
<b>SM_CXDOUBLECLK</b> 36	The width of the rectangle around the location of a first click in a double-click sequence, in pixels. The second click must occur within the rectangle that is defined by SM_CXDOUBLECLK and SM_CYDOUBLECLK for the system to consider the two clicks a double-click. The two clicks must also occur within a specified time. To set the width of the double-click rectangle, call <a href="#">SystemParametersInfo</a> with SPI_SETDOUBLECLKWIDTH.
<b>SM_CXDRAG</b> 68	The number of pixels on either side of a mouse-down point that the mouse pointer can move before a drag operation begins. This allows the user to click and release the mouse button easily without unintentionally starting a drag operation. If this value is negative, it is subtracted from the left of the mouse-down point and added to the right of it.
<b>SM_CXEDGE</b> 45	The width of a 3-D border, in pixels. This metric is the 3-D counterpart of SM_CXBORDER.
<b>SM_CXFIXEDFRAME</b> 7	The thickness of the frame around the perimeter of a window that has a caption but is not sizable, in pixels. SM_CXFIXEDFRAME is the height of the horizontal border, and SM_CYFIXEDFRAME is the width of the vertical border. This value is the same as SM_CXDLGFRAME.
<b>SM_CXFOCUSBORDER</b> 83	The width of the left and right edges of the focus rectangle that the <a href="#">DrawFocusRect</a> draws. This value is in pixels. <b>Windows 2000:</b> This value is not supported.
<b>SM_CXFRAME</b> 32	This value is the same as SM_CXSIZEFRAME.

<b>SM_CXFULLSCREEN</b> 16	The width of the client area for a full-screen window on the primary display monitor, in pixels. To get the coordinates of the portion of the screen that is not obscured by the system taskbar or by application desktop toolbars, call the <a href="#">SystemParametersInfo</a> function with the SPI_GETWORKAREA value.
<b>SM_CXHSCROLL</b> 21	The width of the arrow bitmap on a horizontal scroll bar, in pixels.
<b>SM_CXHTHUMB</b> 10	The width of the thumb box in a horizontal scroll bar, in pixels.
<b>SM_CXICON</b> 11	The default width of an icon, in pixels. The <a href="#">LoadIcon</a> function can load only icons with the dimensions that SM_CXICON and SM_CYICON specifies.
<b>SM_CXICONSPACING</b> 38	The width of a grid cell for items in large icon view, in pixels. Each item fits into a rectangle of size SM_CXICONSPACING by SM_CYICONSPACING when arranged. This value is always greater than or equal to SM_CXICON.
<b>SM_CXMAXIMIZED</b> 61	The default width, in pixels, of a maximized top-level window on the primary display monitor.
<b>SM_CXMAXTRACK</b> 59	The default maximum width of a window that has a caption and sizing borders, in pixels. This metric refers to the entire desktop. The user cannot drag the window frame to a size larger than these dimensions. A window can override this value by processing the <a href="#">WM_GETMINMAXINFO</a> message.
<b>SM_CXMENUCHECK</b> 71	The width of the default menu check-mark bitmap, in pixels.
<b>SM_CXMENUSIZE</b> 54	The width of menu bar buttons, such as the child window close button that is used in the multiple document interface, in pixels.
<b>SM_CXMIN</b> 28	The minimum width of a window, in pixels.
<b>SM_CXMINIMIZED</b> 57	The width of a minimized window, in pixels.
<b>SM_CXMINSIZING</b> 47	The width of a grid cell for a minimized window, in pixels. Each minimized window fits into a rectangle this size when arranged. This value is always greater than or equal to SM_CXMINIMIZED.
<b>SM_CXMINTRACK</b> 34	The minimum tracking width of a window, in pixels. The user cannot drag the window frame to a size smaller than these dimensions. A window can override this value by processing the <a href="#">WM_GETMINMAXINFO</a> message.
<b>SM_CXPADDED BORDER</b> 92	The amount of border padding for captioned windows, in pixels. <b>Windows XP/2000:</b> This value is not supported.

<b>SM_CXSCREEN</b> 0	The width of the screen of the primary display monitor, in pixels. This is the same value obtained by calling <a href="#">GetDeviceCaps</a> as follows: <code>GetDeviceCaps(hdcPrimaryMonitor, HORZRES)</code> .
<b>SM_CXSIZE</b> 30	The width of a button in a window caption or title bar, in pixels.
<b>SM_CXSIZEFRAME</b> 32	The thickness of the sizing border around the perimeter of a window that can be resized, in pixels. SM_CXSIZEFRAME is the width of the horizontal border, and SM_CYSIZEFRAME is the height of the vertical border. This value is the same as SM_CXFRAME.
<b>SM_CXSMICON</b> 49	The recommended width of a small icon, in pixels. Small icons typically appear in window captions and in small icon view.
<b>SM_CXSMSIZE</b> 52	The width of small caption buttons, in pixels.
<b>SM_CXVIRTUALSCREEN</b> 78	The width of the virtual screen, in pixels. The virtual screen is the bounding rectangle of all display monitors. The SM_XVIRTUALSCREEN metric is the coordinates for the left side of the virtual screen.
<b>SM_CXVSCROLL</b> 2	The width of a vertical scroll bar, in pixels.
<b>SM_CYBORDER</b> 6	The height of a window border, in pixels. This is equivalent to the SM_CYEDGE value for windows with the 3-D look.
<b>SM_CYCAPTION</b> 4	The height of a caption area, in pixels.
<b>SM_CYCURSOR</b> 14	The height of a cursor, in pixels. The system cannot create cursors of other sizes.
<b>SM_CYDLGFRAME</b> 8	This value is the same as SM_CYFIXEDFRAME.
<b>SM_CYDOUBLECLK</b> 37	The height of the rectangle around the location of a first click in a double-click sequence, in pixels. The second click must occur within the rectangle defined by SM_CXDOUBLECLK and SM_CYDOUBLECLK for the system to consider the two clicks a double-click. The two clicks must also occur within a specified time. To set the height of the double-click rectangle, call <a href="#">SystemParametersInfo</a> with SPI_SETDOUBLECLKHEIGHT.

<b>SM_CYDRAG</b> 69	The number of pixels above and below a mouse-down point that the mouse pointer can move before a drag operation begins. This allows the user to click and release the mouse button easily without unintentionally starting a drag operation. If this value is negative, it is subtracted from above the mouse-down point and added below it.
<b>SM_CYEDGE</b> 46	The height of a 3-D border, in pixels. This is the 3-D counterpart of SM_CYBORDER.
<b>SM_CYFIXEDFRAME</b> 8	The thickness of the frame around the perimeter of a window that has a caption but is not sizable, in pixels. SM_CXFIXEDFRAME is the height of the horizontal border, and SM_CYFIXEDFRAME is the width of the vertical border. This value is the same as SM_CYDLGFRAME.
<b>SM_CYFOCUSBORDER</b> 84	The height of the top and bottom edges of the focus rectangle drawn by <a href="#">DrawFocusRect</a> . This value is in pixels. <b>Windows 2000:</b> This value is not supported.
<b>SM_CYFRAME</b> 33	This value is the same as SM_CYSIZEFRAME.
<b>SM_CYFULLSCREEN</b> 17	The height of the client area for a full-screen window on the primary display monitor, in pixels. To get the coordinates of the portion of the screen not obscured by the system taskbar or by application desktop toolbars, call the <a href="#">SystemParametersInfo</a> function with the SPI_GETWORKAREA value.
<b>SM_CYHSCROLL</b> 3	The height of a horizontal scroll bar, in pixels.
<b>SM_CYICON</b> 12	The default height of an icon, in pixels. The <a href="#">LoadIcon</a> function can load only icons with the dimensions SM_CXICON and SM_CYICON.
<b>SM_CYICONSPACING</b> 39	The height of a grid cell for items in large icon view, in pixels. Each item fits into a rectangle of size SM_CXICONSPACING by SM_CYICONSPACING when arranged. This value is always greater than or equal to SM_CYICON.
<b>SM_CYKANJIWINDOW</b> 18	For double byte character set versions of the system, this is the height of the Kanji window at the bottom of the screen, in pixels.
<b>SM_CYMAXIMIZED</b> 62	The default height, in pixels, of a maximized top-level window on the primary display monitor.
<b>SM_CYMAXTRACK</b> 60	The default maximum height of a window that has a caption and sizing borders, in pixels. This metric refers to the entire desktop. The user cannot drag the window frame to a size larger than these dimensions. A window can override this value by processing the <a href="#">WM_GETMINMAXINFO</a> message.

<b>SM_CYMENU</b> 15	The height of a single-line menu bar, in pixels.
<b>SM_CYMENUCHECK</b> 72	The height of the default menu check-mark bitmap, in pixels.
<b>SM_CYMENUSIZE</b> 55	The height of menu bar buttons, such as the child window close button that is used in the multiple document interface, in pixels.
<b>SM_CYMIN</b> 29	The minimum height of a window, in pixels.
<b>SM_CYMINIMIZED</b> 58	The height of a minimized window, in pixels.
<b>SM_CYMINSPPACING</b> 48	The height of a grid cell for a minimized window, in pixels. Each minimized window fits into a rectangle this size when arranged. This value is always greater than or equal to SM_CYMINIMIZED.
<b>SM_CYMINTRACK</b> 35	The minimum tracking height of a window, in pixels. The user cannot drag the window frame to a size smaller than these dimensions. A window can override this value by processing the <u>WM_GETMINMAXINFO</u> message.
<b>SM_CYSCREEN</b> 1	The height of the screen of the primary display monitor, in pixels. This is the same value obtained by calling <u>GetDeviceCaps</u> as follows: <code>GetDeviceCaps(hdcPrimaryMonitor, VERTRES)</code> .
<b>SM_CYSIZE</b> 31	The height of a button in a window caption or title bar, in pixels.
<b>SM_CYSIZEFRAME</b> 33	The thickness of the sizing border around the perimeter of a window that can be resized, in pixels. SM_CXSIZEFRAME is the width of the horizontal border, and SM_CYSIZEFRAME is the height of the vertical border. This value is the same as SM_CYFRAME.
<b>SM_CYSMCAPTION</b> 51	The height of a small caption, in pixels.
<b>SM_CYSMICON</b> 50	The recommended height of a small icon, in pixels. Small icons typically appear in window captions and in small icon view.
<b>SM_CYSMSIZE</b> 53	The height of small caption buttons, in pixels.
<b>SM_CYVIRTUALSCREEN</b> 79	The height of the virtual screen, in pixels. The virtual screen is the bounding rectangle of all display monitors. The SM_YVIRTUALSCREEN metric is the coordinates for the top of the virtual screen.

<b>SM_CYVSCROLL</b> 20	The height of the arrow bitmap on a vertical scroll bar, in pixels.
<b>SM_CYVTHUMB</b> 9	The height of the thumb box in a vertical scroll bar, in pixels.
<b>SM_DBCSENABLED</b> 42	Nonzero if User32.dll supports DBCS; otherwise, 0.
<b>SM_DEBUG</b> 22	Nonzero if the debug version of User.exe is installed; otherwise, 0.
<b>SM_DIGITIZER</b> 94	Nonzero if the current operating system is Windows 7 or Windows Server 2008 R2 and the Tablet PC Input service is started; otherwise, 0. The return value is a bitmask that specifies the type of digitizer input supported by the device. For more information, see Remarks. <b>Windows Server 2008, Windows Vista and Windows XP/2000:</b> This value is not supported.
<b>SM_IMMENABLED</b> 82	Nonzero if Input Method Manager/Input Method Editor features are enabled; otherwise, 0. SM_IMMENABLED indicates whether the system is ready to use a Unicode-based IME on a Unicode application. To ensure that a language-dependent IME works, check SM_DBCSENABLED and the system ANSI code page. Otherwise the ANSI-to-Unicode conversion may not be performed correctly, or some components like fonts or registry settings may not be present.
<b>SM_MAXIMUMTOUCHES</b> 95	Nonzero if there are digitizers in the system; otherwise, 0. SM_MAXIMUMTOUCHES returns the aggregate maximum of the maximum number of contacts supported by every digitizer in the system. If the system has only single-touch digitizers, the return value is 1. If the system has multi-touch digitizers, the return value is the number of simultaneous contacts the hardware can provide. <b>Windows Server 2008, Windows Vista and Windows XP/2000:</b> This value is not supported.
<b>SM_MEDIACENTER</b> 87	Nonzero if the current operating system is the Windows XP, Media Center Edition, 0 if not.
<b>SM_MENUDROPALIGNMENT</b> 40	Nonzero if drop-down menus are right-aligned with the corresponding menu-bar item; 0 if the menus are left-aligned.
<b>SM_MIDEASTENABLED</b> 74	Nonzero if the system is enabled for Hebrew and Arabic languages, 0 if not.

<b>SM_MOUSEPRESENT</b> 19	Nonzero if a mouse is installed; otherwise, 0. This value is rarely zero, because of support for virtual mice and because some systems detect the presence of the port instead of the presence of a mouse.
<b>SM_MOUSEHORIZONTALWHEELPRESENT</b> 91	Nonzero if a mouse with a horizontal scroll wheel is installed; otherwise 0.
<b>SM_MOUSEWHEELPRESENT</b> 75	Nonzero if a mouse with a vertical scroll wheel is installed; otherwise 0.
<b>SM_NETWORK</b> 63	The least significant bit is set if a network is present; otherwise, it is cleared. The other bits are reserved for future use.
<b>SM_PENWINDOWS</b> 41	Nonzero if the Microsoft Windows for Pen computing extensions are installed; zero otherwise.
<b>SM_REMOTECONTROL</b> 0x2001	This system metric is used in a Terminal Services environment to determine if the current Terminal Server session is being remotely controlled. Its value is nonzero if the current session is remotely controlled; otherwise, 0. You can use terminal services management tools such as Terminal Services Manager (tsadmin.msc) and shadow.exe to control a remote session. When a session is being remotely controlled, another user can view the contents of that session and potentially interact with it.
<b>SM_REMOTESESSION</b> 0x1000	This system metric is used in a Terminal Services environment. If the calling process is associated with a Terminal Services client session, the return value is nonzero. If the calling process is associated with the Terminal Services console session, the return value is 0. <b>Windows Server 2003 and Windows XP:</b> The console session is not necessarily the physical console. For more information, see <a href="#">WTSGetActiveConsoleSessionId</a> .
<b>SM_SAMEDISPLAYFORMAT</b> 81	Nonzero if all the display monitors have the same color format, otherwise, 0. Two displays can have the same bit depth, but different color formats. For example, the red, green, and blue pixels can be encoded with different numbers of bits, or those bits can be located in different places in a pixel color value.
<b>SM_SECURE</b> 44	This system metric should be ignored; it always returns 0.
<b>SM_SERVERR2</b> 89	The build number if the system is Windows Server 2003 R2; otherwise, 0.
<b>SM_SHOWSOUNDS</b> 70	Nonzero if the user requires an application to present information visually in situations where it would otherwise present the information only in audible form; otherwise, 0.



<b>SM_SHUTTINGDOWN</b> <b>0x2000</b>	Nonzero if the current session is shutting down; otherwise, 0. <b>Windows 2000:</b> This value is not supported.
<b>SM_SLOWMACHINE</b> <b>73</b>	Nonzero if the computer has a low-end (slow) processor; otherwise, 0.
<b>SM_STARTER</b> <b>88</b>	Nonzero if the current operating system is Windows 7 Starter Edition, Windows Vista Starter, or Windows XP Starter Edition; otherwise, 0.
<b>SM_SWAPBUTTON</b> <b>23</b>	Nonzero if the meanings of the left and right mouse buttons are swapped; otherwise, 0.
<b>SM_SYSTEMDOCKED</b> <b>0x2004</b>	Reflects the state of the docking mode, 0 for Undocked Mode and non-zero otherwise. When this system metric changes, the system sends a broadcast message via <a href="#">WM_SETTINGCHANGE</a> with "SystemDockMode" in the LPARAM.
<b>SM_TABLETPC</b> <b>86</b>	Nonzero if the current operating system is the Windows XP Tablet PC edition or if the current operating system is Windows Vista or Windows 7 and the Tablet PC Input service is started; otherwise, 0. The SM_DIGITIZER setting indicates the type of digitizer input supported by a device running Windows 7 or Windows Server 2008 R2. For more information, see Remarks.
<b>SM_XVIRTUALSCREEN</b> <b>76</b>	The coordinates for the left side of the virtual screen. The virtual screen is the bounding rectangle of all display monitors. The SM_CXVIRTUALSCREEN metric is the width of the virtual screen.
<b>SM_YVIRTUALSCREEN</b> <b>77</b>	The coordinates for the top of the virtual screen. The virtual screen is the bounding rectangle of all display monitors. The SM_CYVIRTUALSCREEN metric is the height of the virtual screen.

## Return Value

Type: **Type: int**

If the function succeeds, the return value is the requested system metric or configuration setting.

If the function fails, the return value is 0. [GetLastError](#) does not provide extended error information.

System metrics can vary from display to display.

**GetSystemMetrics**(SM\_CMONITORS) counts only visible display monitors. This is different from [EnumDisplayMonitors](#), which enumerates both visible display monitors and invisible pseudo-monitors that are associated with mirroring drivers. An invisible pseudo-monitor is associated with a pseudo-device used to mirror application drawing for remoting or other purposes.

The SM\_ARRANGE setting specifies how the system arranges minimized windows, and

consists of a starting position and a direction. The starting position can be one of the following values.

Value	Meaning
ARW_BOTTOMLEFT	Start at the lower-left corner of the screen. The default position.
ARW_BOTTOMRIGHT	Start at the lower-right corner of the screen. Equivalent to ARW_STARTRIGHT.
ARW_TOPLEFT	Start at the upper-left corner of the screen. Equivalent to ARW_STARTTOP.
ARW_TOPRIGHT	Start at the upper-right corner of the screen. Equivalent to ARW_STARTTOP   SRW_STARTRIGHT.

The direction in which to arrange minimized windows can be one of the following values.

Value	Meaning
ARW_DOWN	Arrange vertically, top to bottom.
ARW_HIDE	Hide minimized windows by moving them off the visible area of the screen.
ARW_LEFT	Arrange horizontally, left to right.
ARW_RIGHT	Arrange horizontally, right to left.
ARW_UP	Arrange vertically, bottom to top.

The SM\_DIGITIZER setting specifies the type of digitizers that are installed on a device running Windows 7 or Windows Server 2008 R2. The return value is a bitmask that specifies one or more of the following values.

Value	Meaning
NID_INTEGRATED_TOUCH 0x01	The device has an integrated touch digitizer.
NID_EXTERNAL_TOUCH 0x02	The device has an external touch digitizer.
NID_INTEGRATED_PEN 0x04	The device has an integrated pen digitizer.
NID_EXTERNAL_PEN 0x08	The device has an external pen digitizer.
NID_MULTI_INPUT 0x40	The device supports multiple sources of digitizer input.
NID_READY 0x80	The device is ready to receive digitizer input.

This API is not DPI aware, and should not be used if the calling thread is per-monitor DPI aware. For the DPI-aware version of this API, see [GetSystemMetricsForDPI](#). For more information on DPI awareness, see [the Windows High DPI documentation](#).

## Examples

---

The following example uses the [GetSystemMetrics](#) function to determine whether a mouse is installed and whether the mouse buttons are swapped. The example also uses the [SystemParametersInfo](#) function to retrieve the mouse threshold and speed. It displays the information in the console.

```
#include <windows.h>
#include <stdio.h>
#pragma comment(lib, "user32.lib")

void main()
{
    BOOL fResult;
    int aMouseInfo[3];

    fResult = GetSystemMetrics(SM_MOUSEPRESENT);

    if (fResult == 0)
        printf("No mouse installed.\n");
    else
    {
        printf("Mouse installed.\n");

        // Determine whether the buttons are swapped.

        fResult = GetSystemMetrics(SM_SWAPBUTTON);

        if (fResult == 0)
            printf("Buttons not swapped.\n");
        else printf("Buttons swapped.\n");

        // Get the mouse speed and the threshold values.

        fResult = SystemParametersInfo(
            SPI_GETMOUSE, // get mouse information
            0,           // not used
            &aMouseInfo, // holds mouse information
            0);          // not used

        if( fResult )
        {
            printf("Speed: %d\n", aMouseInfo[2]);
            printf("Threshold (x,y): %d,%d\n",
                aMouseInfo[0], aMouseInfo[1]);
        }
    }
}
```

# Requirements

---

<b>Minimum supported client</b>	Windows 2000 Professional [desktop apps only]
<b>Minimum supported server</b>	Windows 2000 Server [desktop apps only]
<b>Target Platform</b>	Windows
<b>Header</b>	winuser.h (include Windows.h)
<b>Library</b>	User32.lib
<b>DLL</b>	User32.dll

## See Also

---

[EnumDisplayMonitors](#)

[GetSystemMetricsForDPI](#)

[SystemParametersInfo](#)

### Note

The feedback system for this content will be changing soon. Old comments will not be carried over. If content within a comment thread is important to you, please save a copy. For more information on the upcoming change, [we invite you to read our blog post](#)