

The Old New Thing

How does the window manager decide where to place a newly-created window?

26 Nov 2012 7:00 AM

16

RATE THIS
★★★★★

Amit wonders [how Windows chooses where to place a newly-opened window](#) on a multiple-monitor system and gives as an example an application whose monitor choice appears inconsistent.

The easy part is if the application specifies where it wants the window to be. In that case, the window is placed at the requested location. How the application chooses those coordinates is up to the application.

On the other hand, if the application passes `CW_USEDEFAULT`, this means that the application is saying, "I have no opinion where the window should go. Please pick a place for me."

If this is the first top-level window created by the application with `CW_USEDEFAULT` as its position, and the `STARTF_USEPOSITION` flag is set in the `STARTUPINFO`, then use the position provided in the `dwX` and `dwY` members.

Officially, that's all you're going to see in the documentation. Past this point is all implementation detail. I'm providing it here to satisfy your curiosity, but please don't write code that relies on it. (This is, I realize, a meaningless request, but I must go through the motions of making it anyway.)

Okay, now let's dive into the various levels of automatic window positioning the window manager performs. Remember, these algorithms are not contractual and can change at any time. (In fact, they *have* changed in the past.) Just to make it harder to rely on this algorithm, I will not tell you which operating system implements the algorithm described below.

From now on, assume that the application has specified `CW_USEDEFAULT` as its position. Also assume that the window is a top-level window.

First we have to choose a monitor.

- If the window was created with an owner, then the window goes onto the monitor associated with the owner window. This tends to keep related windows together on the same monitor.
- Else, if the process was created by the `ShellExecuteEx` function, and the `SEE_MASK_HMONITOR` flag was passed in the `SHELLEXECUTEINFO` structure, then the window goes onto the specified monitor.
- Else, the window goes on the primary monitor.

Next, we have to choose a location on that monitor.

- If this is the first time we need to choose a default location on a monitor, or if the previous default location is too close to the bottom right corner of the monitor, then act as if the previous default location for the monitor was the upper left corner of the monitor.
- The next default location on a monitor is offset from the previous default location, diagonally down and to the right.
 - The vertical offset is chosen so that the top edge of the new window lines up against the bottom of the previous window's caption.
 - The horizontal offset is chosen so that the left edge of the new window lines up against the right edge of the caption icon of the previous window.

The effect of this algorithm is that if you open a bunch of default-positioned windows on a monitor, they line up in a pretty cascade marching down and to the right, until the cascade goes too far, and then they return to the upper left and resume cascading.

Finally, after choosing a monitor and a location on the monitor, the selected location is adjusted (if possible) so that the window does not span monitors.

And that's it, the default-window-positioning algorithm, as it existed in an

Translate this page

Spanish

Microsoft® Translator

Common Tasks

- [RSS for posts](#)
- [RSS for comments](#)
- [Atom](#)

Search

Search MSDN with Bing

☒ Search this blog
 ☐ Search all blogs

News

Holy cow, [I wrote a book!](#)

Basics

- [Archives](#)
- [Ground Rules](#)
- [Suggestion Box](#)
- [Contact Me](#)
- [Disclaimers and such](#)

Blogroll

[The Daily WTF](#)
[Michael Kaplan](#)
[NT Debugging](#)

Categories

[Code](#)
[History](#)
[Non-Computer](#)
[Other \(Computer\)](#)
[Tips/Support](#)
[All tags](#)

unspecified version of Windows. Remember, this algorithm has been tweaked in the past, and it will get tweaked more in the future, so don't rely on it.

Code

Blog - Comment List MSDN TechNet

Comments



skSdnW <#>

26 Nov 2012 7:25 AM

"Else, the window goes on the primary monitor" why not use the same monitor as the current window or the monitor the mouse is on? I guess both designs have their pros and cons.



henke37 <#>

26 Nov 2012 7:29 AM

The minimum version the algorithm could work for is the first version that supports multiple monitors.



Chris B <#>

26 Nov 2012 7:35 AM

I assume this means that SQL Server Management Studio does not create its Filter Settings dialog with an owner as it is consistently positioned on my primary monitor. Which in turn results in me being confused as to why the dialog didn't appear for a few seconds since I don't see it pop in my periphery.



Miff <#>

26 Nov 2012 7:46 AM

@WndSks I'd assume that this is something they changed in a newer version of Windows then the mystery version mentioned here.



Sam <#>

26 Nov 2012 7:51 AM

Speaking of poor multi-monitor support, why can't we have the Start Screen stay up on one monitor while we work with apps on another? Instead the screen disappears whether you like it or no. What gives?



Sam <#>

26 Nov 2012 7:51 AM

Err, I was speaking of Windows 8, of course.



Joshua <#>

26 Nov 2012 8:30 AM

[The next default location on a monitor is offset from the previous default location, diagonally down and to the right.]

I wrote code once that depended on that once. The effect of it failing (which it did whenever it reached the bottom) wasn't bad though.



Kevin <#>

26 Nov 2012 9:06 AM

Ignore the Mac/Windows flamebait comment on the picture, but opening a TON of windows creates some pretty patterns sometimes:

dl.dropbox.com/.../lotsofwindows.png



Mike Caron #

26 Nov 2012 9:32 AM

@Kevin - Sadly, that's not a lot of windows. That's one window being drawn repeatedly on another window who is actively ignoring requests to redraw itself.

[It's probably ignoring paint requests because it crashed. (Think about it.) -Raymond]



Kevin #

26 Nov 2012 9:55 AM

@Mike Caron - I thought this was what happened when the background window was non-responsive, and a process kept crashing. A new crash dialog would kill the old dialog, and create a new one. The new one would get positioned to offset the old window before the old window was destroyed.

I'm aware you can create something similar by dragging a window around and leaving trails behind, but I've also seen exactly this happen completely on its own, which is what I was assuming happened there.



Anonymous Coward #

26 Nov 2012 11:12 AM

Kevin, you can tell this is a dragged-around window by the curvy, irregular, draggy trail like positions of the window images.

When the crash dialogue itself crashes you usually don't get anything special since the standard crash dialogue is centred, not positioned relatively to some other window. Although about five years ago an application with a custom crash handler managed to create a nice staircase pattern.



Kai #

26 Nov 2012 1:01 PM

You keep mentioning those strange things called windows...surely you mean tiles, don't you? ;)



Jolyon Smith #

26 Nov 2012 5:08 PM

I think you meant pains.

I mean panes.

No, my mistake - I meant pains after all. ;)



Ben Cooke #

26 Nov 2012 8:24 PM

Thinking about the pretty diagonal pattern of many default-open windows reminds me of a time I accidentally triggered the Open action on hundreds of files at once and basically hosed my system for a few minutes while it was busy drawing such pretty patterns with whatever program was responsible for those files. I don't recall which, but since it's the default behavior I was observing I don't suppose it matters.



A. Skrobov #

27 Nov 2012 3:01 AM

@WndSks

> "Else, the window goes on the primary monitor" why not use the same monitor as the current window or the monitor the mouse is on?

What if there's no mouse?

What if there's a mouse physically connected, but not actually used ATM?



Neil <#>

27 Nov 2012 3:12 AM

Console windows seem to have their own rules; compared to other applications I get a different number of windows in the cascade when I try. There also seems to have been an incompatible behaviour change (or bug, as I like to call it) starting with Windows XP whereby a console window refuses to automatically position at (0, 0) instead opening offset at the same position as the next one will. (You can easily demonstrate this by opening a number of command prompts in a row.)
