

# Why can't I get the pixels of a window that isn't visible on screen?

 [devblogs.microsoft.com/oldnewthing/20050624-46](http://devblogs.microsoft.com/oldnewthing/20050624-46)

June 24, 2005



Raymond Chen

If the window isn't visible on the screen, then the pixels simply don't exist. The Windows painting model follows the principle of "Don't save anything you can recalculate". Consider: You have a 640×480 display in 16-color mode. That's 150KB of video memory. Take four copies of Notepad and maximize each one. If each of those copies of Notepad required an offscreen cached bitmap, you would be using 450KB of memory just to retain those bits. If your computer has only 640KB, that doesn't leave much memory for anything else. Think about what programs need to display. In the vast majority of cases, the programs already need to keep track of the information needed to generate the display anyway, and usually in a more compact form. Notepad, for example, has an edit control which in turn holds the text being edited. Displayed or not, the edit control needs to keep track of the text. (It's not like Notepad would use OCR to read the pixels from the screen and convert them back to text when it needs them!) Any program that has a scroll bar needs to manage information that is not visible on screen. It's hardly a stretch to make it manage information that **is** visible.

If you do want to get the pixels from a window that isn't visible on screen, you can try sending it the WM\_PRINT message, though this requires that the window you're sending it to even bothers to support the `WM_PRINT` message. (Many don't.) If you are using Windows XP or later, you can use the PrintWindow function.

Raymond Chen

**Follow**

