

# What is the history of the GetRandomRgn function?

 [devblogs.microsoft.com/oldnewthing/20120618-00](http://devblogs.microsoft.com/oldnewthing/20120618-00)

June 18, 2012



Raymond Chen

An anonymous commenter was curious about how the GetRandomRgn function arrived at its strange name, what the purpose of the third parameter is, and why it is inconsistent between Windows 95 and Windows NT. The sense of word “random” here is hacker slang rather than its formal probabilistic definition, specifically sense 2: “Assorted, undistinguished”, perhaps with a bit of sense 4 sprinkled in: “Not well organized.” (Commenter Gabe suggested that a better name would have been GetSpecificRgn.) Once upon a time, when men were men and Windows was 16-bit, there was an internal function used to communicate between the window manager and GDI in order to set up device contexts. Internally, the region was called the Rao Region, named after Rao Remala, the programmer who invented it, and the function that calculated the Rao Region was rather uncreatively called `ComputeRaoRgn`. When porting to 32-bit Windows, the Windows NT and Windows 95 teams both found that they needed this same internal communication between the window manager and GDI. GDI already has a bunch of functions named `GetXxxRgn`, so instead of writing a separate marshaler for each one, they opted to write a single `GetRandomRgn` function which takes an integer which serves as a function code, specifying which region the caller actually wants. (I suspect the Windows 95 team followed the cue of the Windows NT team, since Windows NT ran into the problem first.) Since this was an internal function, it didn’t matter that the name was a bit cutesy, nor did it matter what coordinate system it used, as long as the window manager and GDI agreed on the name and coordinate system. The Windows 95 team still had a lot of 16-bit code that they needed to be compatible with, so they chose to generate the Rao region the same way that the 16-bit `ComputeRaoRgn` function did it. The Windows NT folks, on the other hand, decided that it was more convenient for them that this internal function use screen coordinates, so that’s what it returns on Windows NT. `GetRandomRgn` isn’t really a function that was designed to be public. It was just an internal helper function that outsiders discovered and relied upon to the point that that it became a compatibility constraint so strong that it turned into a *de facto* documented function. And all the weirdness you see behind it is the weirdness of a function never intended for public consumption.

The introduction of the Desktop Window Manager in Windows Vista changed the way the visible region was managed (since all windows are logically visible even when occluded because their drawing is redirected to an off-screen surface), but the `GetRandomRgn`

function has to keep track of the “visible region” anyway, for compatibility.

Raymond Chen

**Follow**

