Functions that return GDI regions rarely actually return regions

devblogs.microsoft.com/oldnewthing/20140327-00

March 27, 2014

Raymond Chen

For reasons I don't quite understand, the only functions in GDI and USER which create GDI regions are the functions with `Create` in their name, like `CreateRectRgn` or `CreatePolygonRgn`. All the other functions which return a region require you to pass an existing region to use as the output.

I can see this being useful for `CombineRgn`, because you can set your output region to be equal to one of the input regions in order to update a region in place.

```
// hrgnTotal |= hrgnSegment
CombineRgn(hrgnTotal, hrgnTotal, hrgnSegment, RGN_OR);
// hrgnTotal &= hrgnClip
CombineRgn(hrgnTotal, hrgnTotal, hrgnClip, RGN_AND);
```

But for all of the `Get` functions, having to create an output region is usually just an annoyance.

```
// Create a dummy region - contents not important
HRGN hrgnClip = CreateRectRgn(0, 0, 0, 0);
// Ask for the clipping region to be copied to the dummy region
int status = GetClipRgn(hdc, hrgnClip);
```

I guess it lets you reuse a single dummy region over and over again, but in practice, you're just going to destroy the region when you're done to free up the GDI region memory.

Whatever the historical reason for this, we're stuck with it. It may be an ugly pattern, but at least it's a pattern.

The things I do for you: I asked a colleague who worked on Windows 3.0 if he knew the reason for this design pattern. He didn't know but suggested that I ask another person who retired from Microsoft many, many years ago. Fortunately, I happened to have his email address even though we aren't really that close. And the second person also didn't know.

"This behavior was already in place when I joined the Windows 1.03 project. Maybe you can ask Rao." Unfortunately, I don't have Rao's email address, so the trail ran cold. But I burned a favor for you guys.

Raymond Chen

Follow