

What is the correct way of using SaveDC and RestoreDC?

devblogs.microsoft.com/oldnewthing/20170920-00

September 20, 2017



Raymond Chen

There are these two functions `SaveDC` and `RestoreDC`. How do they work?

Each device context (DC) maintains a stack of saved states. When you call `SaveDC`, the current state of the DC is saved and pushed onto the stack, and you get a positive integer representing that saved state. If you call `SaveDC` again, a new saved state is created and pushed onto the stack, and you get a new integer that represents the second state. Each call to `SaveDC` pushes the current state onto this internal stack.

There are two ways to call the `RestoreDC` function. One is to pass a negative number. This indicates how many states to pop off the stack, and the last state popped off the stack is applied to the DC.

The other (more common) way to call the `RestoreDC` function is to pass a specific state. In that case, the specific state is restored, and that state is popped off the stack. And since it's a stack, this also means that any states that were pushed onto the stack after that point are also popped off.

And of course you cannot restore a state to a DC different from the DC you saved it from. (Because each DC has a separate stack of saved states.)

For concreteness, let's say that we've saved the state three times:

```
SelectObject(hdc, GetStockObject(NULL_BRUSH));
int state1 = SaveDC(hdc);
SelectObject(hdc, GetStockObject(WHITE_BRUSH));
int state2 = SaveDC(hdc);
SelectObject(hdc, GetStockObject(BLACK_BRUSH));
int state3 = SaveDC(hdc);
SelectObject(hdc, GetStockObject(DC_BRUSH));
```

State 1 has the null brush, state 2 has the white brush, state 3 has the black brush, and the current state (not saved) has the DC brush.

Here's what could happen next:

Negative argument	Equivalent positive argument	Result	Stack
RestoreDC(-1)	RestoreDC(state3)	Current brush is black	state2 (top), state1
RestoreDC(-2)	RestoreDC(state2)	Current brush is white	state1
RestoreDC(-3)	RestoreDC(state1)	Current brush is null	empty

Note that the state that is applied gets popped off the stack, which means that each state can be restored at most once.

Raymond Chen

Follow

