

## Memory Device Context

-

[Alyce](#)

[CreateCompatibleDC](#) | [Contents of Memory DC](#) | [DeleteDC](#)

*Some text below is copied from the Microsoft Developers Network Library.*

For an eBook or printed book on using the API with Liberty BASIC, see:  
[APIs for Liberty BASIC](#)

## CreateCompatibleDC

Liberty BASIC allows us to display graphics in a graphicbox control or a window opened for graphics. We cannot draw in memory with native commands. We can draw in memory with GDI commands. This requires us first to create a memory Device Context. We do this with CreateCompatibleDC.

The CreateCompatibleDC function creates a memory device context (DC) compatible with the specified device.

The syntax is as follows.

```
calldll #gdi32, "CreateCompatibleDC", _  
hdc as ulong, _ 'graphicbox DC  
hMemDC as ulong 'memory DC
```

We pass into the function the handle to an existing DC. If this handle is NULL, the function creates a memory DC compatible with the application's current screen. If the function succeeds, the return value is the handle to a memory DC. If it fails, the return value is 0.

Here is a snippet with the function to get the graphicbox DC, followed by the function to create a memory DC.

```
h=hwnd(#1.g) 'graphicbox handle  
  
'get device context for window:  
calldll #user32, "GetDC", _  
h as ulong, _ 'graphicbox handle  
hdc as ulong 'returns handle to device context  
  
calldll #gdi32, "CreateCompatibleDC", _
```

```
hdc as ulong, _ 'graphicbox DC
hMemDC as ulong 'memory DC
```

## Contents of Memory DC

A memory DC exists only in memory. When the memory DC is created, its display surface is exactly one monochrome pixel wide and one monochrome pixel high. Before an application can use a memory DC for drawing operations, it must select a bitmap of the correct width and height into the DC with `SelectObject`. The bitmap can be an image that was loaded into memory with the native `LOADBMP` command or it can be loaded with the API `LoadImageA` or it can be loaded by a third-party DLL. It can also be created in memory with the `CreateCompatibleBitmap` function, which allows us to specify the height, width, and color organization required.

We'll discuss `CreateCompatibleBitmap` in the next tutorial.

## DeleteDC

The `DeleteDC` function deletes the specified device context. The return value is nonzero if the function succeeds.

```
call dll #gdi32, "DeleteDC", _
hMemDC as ulong, _ 'DC to delete
re as long 'nonzero=success
```

---

[GDI Tutorials Home](#)