

What is the mysterious fourth message box button?

 devblogs.microsoft.com/oldnewthing/20081219-00

December 19, 2008



Raymond Chen

When you call the `MessageBox` function, you pass flags specifying which of a fixed set of button patterns you want (for example, Yes/No and OK/Cancel) and which button you want to be the default (`MB_DEFBUTTON1` through `MB_DEFBUTTON4` .)

Wait a second. What's with this `MB_DEFBUTTON4` ? None of the button patterns are four-button patterns. The highest number of buttons you can specify is three: Abort/Retry/Ignore. How can you set a nonexistent button to be the default?

Let's do some header file spelunking. The flag for this magical fourth button is defined here:

```
#define MB_DEFBUTTON1          0x00000000L
#define MB_DEFBUTTON2          0x00000100L
#define MB_DEFBUTTON3          0x00000200L
#if(WINVER >= 0x0400)
#define MB_DEFBUTTON4          0x00000300L
#endif /* WINVER >= 0x0400 */
```

Aha, the magic fourth button was added in `WINVER` 4.0. Therefore, whatever the fourth button is, it was introduced when `WINVER == 0x0400` . Let's see what other message box flags were introduced then:

```

#define MB_OK 0x00000000L
#define MB_OKCANCEL 0x00000001L
#define MB_ABORTRETRYIGNORE 0x00000002L
#define MB_YESNOCANCEL 0x00000003L
#define MB_YESNO 0x00000004L
#define MB_RETRYCANCEL 0x00000005L
#if(WINVER >= 0x0500)
#define MB_CANCELTRYCONTINUE 0x00000006L
#endif /* WINVER >= 0x0500 */

#define MB_ICONHAND 0x00000010L
#define MB_ICONQUESTION 0x00000020L
#define MB_ICONEXCLAMATION 0x00000030L
#define MB_ICONASTERISK 0x00000040L

#if(WINVER >= 0x0400)
#define MB_USERICON 0x00000080L
#define MB_ICONWARNING MB_ICONEXCLAMATION
#define MB_ICONERROR MB_ICONHAND
#endif /* WINVER >= 0x0400 */

#define MB_ICONINFORMATION MB_ICONASTERISK
#define MB_ICONSTOP MB_ICONHAND

#define MB_DEFBUTTON1 0x00000000L
#define MB_DEFBUTTON2 0x00000100L
#define MB_DEFBUTTON3 0x00000200L
#if(WINVER >= 0x0400)
#define MB_DEFBUTTON4 0x00000300L
#endif /* WINVER >= 0x0400 */

#define MB_APPLMODAL 0x00000000L
#define MB_SYSTEMMODAL 0x00001000L
#define MB_TASKMODAL 0x00002000L
#if(WINVER >= 0x0400)
#define MB_HELP 0x00004000L // Help Button
#endif /* WINVER >= 0x0400 */

#define MB_NOFOCUS 0x00008000L
#define MB_SETFOREGROUND 0x00010000L
#define MB_DEFAULT_DESKTOP_ONLY 0x00020000L

#if(WINVER >= 0x0400)
#define MB_TOPMOST 0x00040000L
#define MB_RIGHT 0x00080000L

```

```

#define MB_RTLCREADING                0x00100000L
#endif /* WINVER >= 0x0400 */

#ifdef _WIN32_WINNT
#if (_WIN32_WINNT >= 0x0400)
#define MB_SERVICE_NOTIFICATION      0x00200000L
#else
#define MB_SERVICE_NOTIFICATION      0x00040000L
#endif
#define MB_SERVICE_NOTIFICATION_NT3X 0x00040000L
#endif

```

We can discount the flags like `MB_ICONWARNING` which are just alternate names for existing flags, as well as `MB_SERVICE_NOTIFICATION` which already existed but with a different value. This leaves the following:

```

#define MB_USERICON                    0x00000080L
#define MB_HELP                        0x00004000L // Help Button
#define MB_TOPMOST                     0x00040000L
#define MB_RIGHT                       0x00080000L
#define MB_RTLCREADING                 0x00100000L

```

Of these flags, `MB_USERICON` affects the icon, and `MB_TOPMOST`, `MB_RIGHT` and `MB_RTLCREADING` affect the dialog box's position and layout; none of them affect the buttons. But wait, there's `MB_HELP`. Ah, that flag "adds a **Help** button to the message box." That's our magical fourth button! Let's celebrate by showing a four-button message box with the default set to the fourth button:

```

#include <windows.h>

int WINAPI WinMain(HINSTANCE hinst, HINSTANCE hinstPrev,
                  LPSTR lpCmdLine, int nShowCmd)
{
    return MessageBox(NULL, TEXT("Four buttons!"), TEXT("Title"),
                     MB_ABORTRETRYIGNORE | MB_HELP | MB_DEFBUTTON4);
}

```

[Raymond Chen](#)

Follow

