Windows Function Calls

Windows Function Calls in 2023

The Windows MessageBox function is still widely used today, and it is still defined in the same way as it was in the 16-bit versions of Windows.

However, there are a few important things to keep in mind when using MessageBox in 2023:

1. All **Windows programs should be compiled with Unicode support.** This means that you should define the UNICODE identifier in your program's preprocessor definitions.

2. When you use MessageBox in your program, the compiler will generate a call to MessageBoxW if the UNICODE identifier is defined, and a call to MessageBoxA if it is not defined.

3. The **MessageBoxA function** is still available in Windows 10 and newer, but it is not recommended for use in new programs.

Here is an example of how to use MessageBox in a modern Windows program:

```
#include <windows.h>

int main ()
{
    int result = MessageBox(NUll, "This is a messagebox.", "My Program", MB_OK);
    if(result == IDOK)
    {
        //User clicked ok...do something.
    }
    return 0;
}
```

This code will display a message box with the title "My Program" and the message "This is a message box." The user will have the option to click OK to close the message box.

Coding

Here is the code in **WINUSER.H** that does the trick:

```
#ifdef UNICODE
#define MessageBox MessageBoxW
#else
#define MessageBox MessageBoxA
#endif
```

This **code checks if the UNICODE identifier is defined**. If it is, then the MessageBox macro is defined to be MessageBoxW. Otherwise, the MessageBox macro is defined to be MessageBoxA.

This means that all calls to the MessageBox function in your program will actually be **calls to MessageBoxW** if the UNICODE identifier is defined, and **calls to MessageBoxA** if it is not defined.

Conclusion

When using the **MessageBox function** in 2023, it is important to keep in mind the following:

All Windows programs should be compiled with Unicode support.

The **MessageBoxA function** is still available, but it is **not recommended** for use in new programs.

You can use the MessageBox macro to ensure that your program calls the correct version of the MessageBox function, regardless of whether Unicode support is enabled.