

How do I get access to the `wParam` and `lParam` of the `WM_QUERYENDSESSION` method from my MFC message handler?

 devblogs.microsoft.com/oldnewthing/20231221-00

December 21, 2023



Raymond Chen

The Microsoft Foundation Classes (MFC) framework provides a set of macros for building message handler tables. The handler for the `WM_QUERYENDSESSION` message has the signature

```
BOOL OnQueryEndSession();
```

and it does not have access to the `wParam` and `lParam` parameters.¹ How can you get the values of those parameters?

Don't be afraid to learn how things work. That often leads to insights on how to use them better.

In our case, `ON_WM_QUERYENDSESSION` goes like this:

```
#define ON_WM_QUERYENDSESSION() \
    { WM_QUERYENDSESSION, 0, 0, 0, AfxSig_bv, \
      (AFX_PMSG)(AFX_PMSGW)(BOOL (AFX_MSG_CALL CWnd::*)(void)) \
      &OnQueryEndSession },
```

If you read the comment block at the top of the file, it says that message map entries can take many forms, including

```
7) constant windows messages
    nMessage, 0, 0, 0, signature type, member function
    (eg: WM_PAINT, 0, ...)
```

From the structure of the table and the pattern of the macros, and the internal documentation, it's apparent that the message map is a table of message number and instructions on how to handle the message. The signature `AfxSig_bv` is documented in the header file as

```
Afx_bv = Afx_wv,    // BOOL (void)
```

Apparently, what happens is that the message dispatcher looks through the table for a matching message number, and once it finds a match, it calls the associated function using the specified signature, and then returns the result as the message result.

We can therefore just create our own entry for `WM_QUERYENDSESSION` that uses a different signature. And it turns out that the `ON_MESSAGE` macro exists specifically for generic messages:

```
#define ON_MESSAGE(message, memberFxn) \  
    { message, 0, 0, 0, AfxSig_lwl, \  
      (AFX_PMSG)(AFX_PMSGW)(BOOL (AFX_MSG_CALL CWnd::*)(WPARAM, LPARAM)) \  
      &memberFxn },
```

Solution: Treat `WM_QUERYENDSESSION` as a custom message:

```
ON_MESSAGE(WM_QUERYENDSESSION, OnQueryEndSession)  
  
LRESULT AFX_MSG_CALL OnQueryEndSession(WPARAM wParam, LPARAM lParam);
```

Bonus chatter: There apparently used to be a `CWnd::GetCurrentMessage()` function for getting a copy of the current message being handled by MFC. (I don't know this for sure, but I see references to it.) It doesn't exist anymore (or maybe never existed), so it's not really an option.

¹ The reason is that at the time that MFC was written, there was no information encoded in those parameters.