Parameter	Type/Description
lplpfnNextHook	**FARPROC FAR *** Points to a memory location that contains the **FARPROC** structure returned by the **SetWindowsHook** function. Windows changes the value at this location after an application calls the **UnhookWindowsHook** function.

Return Value The return value specifies a value that is directly related to the *code* parameter.

DefineHandleTable 3.0

Syntax **BOOL DefineHandleTable(***wOffset***)**

This function creates a private handle table in an application's default data segment. The application stores in the table the segment addresses of global memory objects returned by the **GlobalLock** function. In real mode, Windows updates the corresponding address in the private handle table when it moves a global memory object. When Windows discards an object with a corresponding table entry, Windows replaces the address of the object in the table with the object's handle. Windows does not update addresses in the private handle table in protected (standard or 386 enhanced) mode.

Parameter	Description
wOffset	**WORD** Specifies the offset from the beginning of the data segment to the beginning of the private handle table. If *wOffset* is zero, Windows no longer updates the private handle table.

Return Value The return value is nonzero if the function was successful. Otherwise, it is zero.

Comments The private handle table has the following format:

```
Count
Clear_Number
Entry[0]
.
.
.
Entry[Count-1]
```

The first **WORD** (*Count*) in the table specifies the number of entries in the table. The second **WORD** (*Clear_Number*) specifies the number of entries (from the beginning of the table) which Windows will set to zero when Windows updates its least-recently-used

(LRU) memory list. The remainder of the table consists of an array of addresses returned by **GlobalLock**.

The application must initialize the *Count* field in the table before calling **DefineHandleTable**. The application can change either the *Count* or *Clearn_Number* fields at any time.

DefMDIChildProc 3.0

Syntax **LONG DefMDIChildProc(***hWnd, wMsg, wParam, lParam***)**

This function provides default processing for any Windows messages that the window function of a multiple document interface (MDI) child window does not process. All window messages that are not explicitly processed by the window function must be passed to the **DefMDIChildProc** function, not the **DefWindowProc** function.

Parameter	Type/Description	
hWnd	**HWND**	Identifies the MDI child window.
wMsg	**WORD**	Specifies the message number.
wParam	**WORD**	Specifies 16 bits of additional message-dependent information.
lParam	**DWORD**	Specifies 32 bits of additional message-dependent information.

Return Value The return value specifies the result of the message processing and depends on the actual message sent.

Comments This function assumes that the parent of the window identified by the *hWnd* parameter was created with the MDICLIENT class.

The source code for the **DefMDIChildProc** function is provided on the SDK distribution disks.

Normally, when an application's window procedure does not handle a message, it passes the message to the **DefWindowProc** function, which processes the message. MDI applications use the **DefFrameProc** and **DefMDIChildProc** functions instead of **DefWindowProc** to provide default message processing. All messages that an application would normally pass to **DefWindowProc** (such as nonclient messages and WM_SETTEXT) should be passed to **DefMDIChildProc** instead. In addition to these, **DefMDIChildProc** also handles the following messages:

Message	Default Processing by DefMDIChildProc