Checking for Errors

In Windows API programming, it's crucial to check for errors when using various API functions. If an API function returns an error value (typically NULL), you can call the GetLastError API function to obtain more information about the error. GetLastError returns a 32-bit integer error code in the EAX register.

GetLastError Function:
GetLastError is used to retrieve error information after a Windows API function call. It returns a 32-bit integer error code in the EAX register, which can be used to identify the specific error.

If a function returns NULL or an error code, you can call GetLastError to get more details about what went wrong.

FormatMessage Function:
After obtaining the error code from GetLastError, you might want to retrieve a human-readable error message to better understand the error.

FormatMessage is used for this purpose. It formats a message based on the error code. Its parameters are:

dwFlags: Formatting options, specifying how to interpret the lpSource parameter. Recommended values include FORMAT_MESSAGE_ALLOCATE_BUFFER and FORMAT_MESSAGE_FROM_SYSTEM.

lpSource: Location of the message definition. For system error messages, set it to NULL (0).

dwMsgID: The error code obtained from GetLastError.

dwLanguageID: Language identifier. Set to zero for a language-neutral or user's default locale message.

lpBuffer (output parameter): Pointer to a buffer that receives the null-terminated message string. If FORMAT_MESSAGE_ALLOCATE_BUFFER is used, the buffer is allocated automatically.

nSize: Buffer size, which can be set to 0 if using the recommended dwFlags options. va_list: Pointer to an array of values that can be inserted in a formatted message. Not used for error messages.

Sample Use of GetLastError and FormatMessage:

First, you call GetLastError to obtain the error code. Then, you invoke FormatMessage to retrieve the corresponding error message based on the error code. Finally, you can display or handle the error message as needed.

LocalFree Function:

After using FormatMessage to obtain the error message, it's important to release the storage allocated by FormatMessage.

You can use LocalFree for this purpose. The code provided demonstrates the use of GetLastError and FormatMessage to obtain and display error messages in a Windows API program.

It checks for errors, retrieves the error message, and frees allocated memory after use.

Note: Error codes and messages are essential for debugging and providing meaningful feedback to users when errors occur in Windows API applications.

```
.data
messageId DWORD ?
pErrorMsg DWORD ? ; points to error message

.code
call GetLastError
mov messageId, eax

INVOKE FormatMessage, FORMAT_MESSAGE_ALLOCATE_BUFFER +
FORMAT_MESSAGE_FROM_SYSTEM, NULL, messageId, 0, ADDR pErrorMsg, 0,
NULL

; After calling FormatMessage, you can use the error message in
pErrorMsg

; ... (perform error handling or display the error message as needed)

; Don't forget to free the allocated memory when done
INVOKE LocalFree, pErrorMsg
```

This code checks for errors by calling GetLastError, retrieves the error message using FormatMessage, and then handles or displays the error message. Finally, it frees the allocated memory using LocalFree.

```
.data
    messageId DWORD ?
    pErrorMsg DWORD ? ; points to error message
.code
    call GetLastError
    mov messageId, eax

    INVOKE FormatMessage, FORMAT_MESSAGE_ALLOCATE_BUFFER +
FORMAT_MESSAGE_FROM_SYSTEM, NULL, messageId, 0, ADDR pErrorMsg, 0,
NULL

    ; After calling FormatMessage, you can use the error message in
pErrorMsg

    ; ... (perform error handling or display the error message as
needed)

    ; Don't forget to free the allocated memory when done
    INVOKE LocalFree, pErrorMsg
WriteWindowsMsg PROC USES eax edx
; Displays a string containing the most recent error
; generated by MS-Windows.
; Receives: nothing
; Returns: nothing
.data
    WriteWindowsMsg_1 BYTE "Error ",0
    WriteWindowsMsg_2 BYTE ": ",0
    pErrorMsg DWORD ?
    ; points to the error message
    messageId DWORD ?
.code
    ; Get the most recent error code
    call GetLastError
    mov messageId, eax
    ; Display the error number as "Error X: "
    mov edx, OFFSET WriteWindowsMsg_1
    call WriteString
    call WriteDec
    mov edx, OFFSET WriteWindowsMsg_2
    call WriteString
    ; Get the corresponding error message string
    INVOKE FormatMessage, FORMAT_MESSAGE_ALLOCATE_BUFFER + \
    FORMAT_MESSAGE_FROM_SYSTEM, NULL, messageId, 0, ADDR pErrorMsg,
0, 0
    ; Display the error message generated by MS-Windows
    mov edx, pErrorMsg
    call WriteString
    ; Free the error message string
```

```
    INVOKE LocalFree, pErrorMsg
    ret
    WriteWindowsMsg ENDP
```

Explanation:

The WriteWindowsMsg procedure is designed to display error messages generated by the MS-Windows operating system.

It starts by calling GetLastError to obtain the most recent error code and stores it in the messageId variable.

The procedure then displays the error number in the format "Error X: " where X is the error code.

It uses the WriteString procedure to display the "Error " and the WriteDec procedure to display the error code.

After that, it retrieves the corresponding error message string using FormatMessage. The FORMAT_MESSAGE_ALLOCATE_BUFFER and FORMAT_MESSAGE_FROM_SYSTEM flags are used to allocate memory for the message and obtain it from the system.

The obtained error message is displayed using WriteString.

Finally, it frees the memory allocated for the error message using LocalFree.

This procedure is a convenient way to retrieve and display error messages when working with Windows API functions, making it easier to diagnose issues in your applications.