

Paul King

Message Bar

**Application Programming Interface
Version 1.0**

December 30, 1992

CONTENTS

μCONTENTS	i
Preface	1
Intended Audience	1
Message Bar API	1
InitMB	2
Function Prototype	2
Purpose2	
Variables	2
Returns	2
Comments	2
KillMB	3
Function Prototype	3
Purpose3	
Variables	3
Comments	3
ResizeMB	4
Function Prototype	4
Purpose4	
Variables	4
Comments	4
ColorMB	5
Function Prototype	5
Purpose5	
Variables	5
Returns	5
Comments	5
MBText	6
Function Prototype	6
Purpose6	
Variables	6
Comments	6
MBSHOW	7
Function Prototype	7
Purpose7	
Variables	7
Comments	7

Preface

This document provides a detailed description of the application programming interface (API) used with the Message Bar DLL (MESSAGEB.DLL). All software interfaces are available from any application that can call routines located in DLL's. All of this software was written in C.

Message Bar is a control that can be used by an application to display messages in the lower portion of the application's main window. This API explains all of the current functionality supported.

Intended Audience

This document is for programmers. The reader should have a thorough understanding of MS-DOS, Microsoft Windows, and C.

Message Bar API

The following function calls are exported in the MESSAGEB.DLL:

- **InitMB**
- **KillMB**
- **ResizeMB**
- **ColorMB**
- **MBText**
- **MBSHOW**

Each of these functions is described below.

InitMB

Function Prototype

InitMB(HWND, HINSTANCE, COLORREF, COLORREF, BOOL)

Purpose

Init the message bar

Variables

HWND

Handle of the Window that will contain the Message Bar.

HINSTANCE

Handle of the Instance of the Owning Application.

COLORREF

Message Bar Background Color

COLORREF

Message Bar Text Color

BOOL

Initial display state

Returns

BOOL

TRUE on success, FALSE on failure.

Comments

InitMB should only be called **ONCE** per application. The current design of MESSAGEB.DLL only allows one message bar per application. The calling program may specify the text and background colors to be displayed on the status bar as well as the initial display state of the status bar. This call should be placed after the CreateWindow call for the main application window.

KillMB

Function Prototype

KillMB(void)

Purpose

Remove the message bar

Variables

NONE

Comments

KillMB must be called **ONCE** for each application that issued and received a TRUE response from a InitMB call. Not calling KillMB will eventually cause Windows to crash do to lack of resources. This call should be placed in the **WM_DESTROY** message of the application callback procedure.

ResizeMB

Function Prototype

ResizeMB(void)

Purpose

Re-sizes the message bar

Variables

NONE

Comments

ResizeMB must be placed in the **WM_SIZE** message of the application callback procedure. If this function call is not placed here, the message bar will not change size as the owning window changes size.

ColorMB

Function Prototype

ColorMB(WPARAM, LPARAM)

Purpose

Paint the Message Bar

Variables

WPARAM

wParam from the **WM_CTLCOLOR** Message.

LPARAM

lParam from the **WM_CTLCOLOR** Message.

Returns

LRESULT

To be returned to Windows.

Comments

This call should be placed in the **WM_CTLCOLOR** message of the application callback procedure. The result should be returned to Windows via the return statement. An example of the use of this call is in **SAMPLE.C**. If this call is not used, the control will never be painted.

MBText

Function Prototype

MBText(LPSTR)

Purpose

Place text on the Message Bar

Variables

LPSTR

Long pointer to a NULL terminated string. There is no guarantee that all of the contents of the string will be displayed. If the string is longer than the display, the string will be clipped. All strings are left-justified within the control.

Comments

This function places text in the Message Bar. To remove the text, call MBText ("").

MBSHOW

Function Prototype

MBSHOW(BOOL)

Purpose

Turn the Message Bar on or off

Variables

BOOL

TRUE turns the Message Bar on. FALSE hides the Message Bar.

Comments