3 ComDlg32.lib

3.1 ChooseColor

The ChooseColor function creates a Color dialog box that enables the user to select a color.

```
ChooseColor: procedure
(
    var lpcc    :CHOOSECOLOR
);
    stdcall;
    returns( "eax" );
    external( "__imp__ChooseColorA@4" );
```

Parameters

lpcc

[in/out] Pointer to a CHOOSECOLOR structure that contains information used to initialize the dialog box. When ChooseColor returns, this structure contains information about the user's color selection.

Return Values

If the user clicks the OK button of the dialog box, the return value is nonzero. The rgbResult member of the CHOOSECOLOR structure contains the RGB color value of the color selected by the user.

If the user cancels or closes the Color dialog box or an error occurs, the return value is zero. To get extended error information, call the CommDlgExtendedError function, which can return one of the following values:

```
CDERR_DIALOGFAILURE
CDERR_FINDRESFAILURE
CDERR_MEMLOCKFAILURE
CDERR_INITIALIZATION
CDERR_NOHINSTANCE
CDERR_LOCKRESFAILURE
CDERR_NOHOOK
CDERR_LOADRESFAILURE
CDERR_NOTEMPLATE
CDERR_LOADSTRFAILURE
CDERR_STRUCTSIZE
CDERR_MEMALLOCFAILURE
```

Remarks

The Color dialog box does not support palettes. The color choices offered by the dialog box are limited to the system colors and dithered versions of those colors.

You can provide a CCHookProc hook procedure for a Color dialog box. The hook procedure can process messages sent to the dialog box. To enable a hook procedure, set the CC_ENABLEHOOK flag in the Flags member of the CHOOSECOLOR structure and specify the address of the hook procedure in the lpfnHook member.

Requirements

Windows NT/2000: Requires Windows NT 3.1 or later.

Windows 95/98: Requires Windows 95 or later.

3.2 ChooseFont

The ChooseFont function creates a Font dialog box that enables the user to choose attributes for a logical font. These attributes include a typeface name, style (bold, italic, or regular), point size, effects (underline, strikeout, and text color), and a script (or character set).

```
ChooseFont: procedure
(
    var lpcf    :CHOOSEFONT
);
    stdcall;
    returns( "eax" );
    external( "__imp__ChooseFontA@4" );
```

Parameters

lpcf

[in/out] Pointer to a CHOOSEFONT structure that contains information used to initialize the dialog box. When ChooseFont returns, this structure contains information about the user's font selection.

Return Values

If the user clicks the OK button of the dialog box, the return value is nonzero. The members of the CHOOSE-FONT structure indicate the user's selections.

If the user cancels or closes the Font dialog box or an error occurs, the return value is zero. To get extended error information, call the CommDlgExtendedError function, which can return one of the following values:

CDERR_DIALOGFAILURE

CDERR_FINDRESFAILURE

CDERR_NOHINSTANCE

CDERR_INITIALIZATION

CDERR_NOHOOK

CDERR_LOCKRESFAILURE

CDERR_NOTEMPLATE

CDERR_LOADRESFAILURE

CDERR_STRUCTSIZE

CDERR_LOADSTRFAILURE

CFERR_MAXLESSTHANMIN

CDERR_MEMALLOCFAILURE

CFERR_NOFONTS

CDERR_MEMLOCKFAILURE

Remarks

You can provide a CFHookProc hook procedure for a Font dialog box. The hook procedure can process messages

sent to the dialog box. To enable a hook procedure, set the CF_ENABLEHOOK flag in the Flags member of the CHOOSEFONT structure and specify the address of the hook procedure in the lpfnHook member.

The hook procedure can send the WM_CHOOSEFONT_GETLOGFONT, WM_CHOOSEFONT_SETFLAGS, and WM_CHOOSEFONT_SETLOGFONT messages to the dialog box to get and set the current values and flags of the dialog box.

Requirements

Windows NT/2000: Requires Windows NT 3.1 or later.

Windows 95/98: Requires Windows 95 or later.

3.3 CommDlgExtendedError

The CommDlgExtendedError function returns a common dialog box error code. This code indicates the most recent error to occur during the execution of one of the common dialog box functions. These error codes are defined in Cderr.h.

```
CommDlgExtendedError: procedure;
    stdcall;
    returns( "eax" );
    external( "__imp__CommDlgExtendedError@0" );
```

Parameters

This function has no parameters.

Return Values

If the most recent call to a common dialog box function succeeded, the return value is undefined.

If the common dialog box function returned FALSE because the user closed or canceled the dialog box, the return value is zero. Otherwise, the return value is a nonzero error code. For more information, see the following Remarks section.

Remarks

The CommDlgExtendedError function can return general error codes for any of the common dialog box functions. In addition, there are error codes that are returned only for a specific common dialog box. The error codes returned by CommDlgExtendedError are defined in the Cderr.h file.

The following general error codes can be returned for any of the common dialog box functions.

Value	Meaning
CDERR_DIALOGFAILURE	The dialog box could not be created. The common dialog box function's call to the DialogBox function failed. For example, this error occurs if the common dialog box call specifies an invalid window handle.
CDERR_FINDRESFAILURE	The common dialog box function failed to find a specified resource.
CDERR_INITIALIZATION	The common dialog box function failed during initialization. This error often occurs when sufficient memory is not available.
CDERR_LOADRESFAILURE	The common dialog box function failed to load a specified resource.

CDERR_LOADSTRFAILURE	The common dialog box function failed to load a specified string.
CDERR_LOCKRESFAILURE	The common dialog box function failed to lock a specified resource.
CDERR_MEMALLOCFAILURE	The common dialog box function was unable to allocate memory for internal structures.
CDERR_MEMLOCKFAILURE	The common dialog box function was unable to lock the memory associated with a handle.
CDERR_NOHINSTANCE	The ENABLETEMPLATE flag was set in the Flags member of the initialization structure for the corresponding common dialog box, but you failed to provide a corresponding instance handle.
CDERR_NOHOOK	The ENABLEHOOK flag was set in the Flags member of the initialization structure for the corresponding common dialog box, but you failed to provide a pointer to a corresponding hook procedure.
CDERR_NOTEMPLATE	The ENABLETEMPLATE flag was set in the Flags member of the initialization structure for the corresponding common dialog box, but you failed to provide a corresponding template.
CDERR_REGISTERMSGFAIL	The RegisterWindowMessage function returned an error code when it was called by the common dialog box function.
CDERR_STRUCTSIZE	The lStructSize member of the initialization structure for the corresponding common dialog box is invalid.

The following error codes can be returned for the PrintDlg function.

Value	Meaning
PDERR_CREATEICFAILURE	The PrintDlg function failed when it attempted to create an information context.
PDERR_DEFAULTDIFFERENT	You called the PrintDlg function with the DN_DEFAULTPRN flag specified in the wDefault member of the DEVNAMES structure, but the printer described by the other structure members did not match the current default printer. (This error occurs when you store the DEVNAMES structure and the user changes the default printer by using the Control Panel.) To use the printer described by the DEVNAMES structure, clear the DN_DEFAULTPRN flag and call PrintDlg again. To use the default printer, replace the DEVNAMES structure (and the DEVMODE structure, if one exists) with NULL; and call PrintDlg again.
PDERR_DNDMMISMATCH	The data in the DEVMODE and DEVNAMES structures describes two different printers.
PDERR_GETDEVMODEFAIL	The printer driver failed to initialize a DEVMODE structure. (This error code applies only to printer drivers written for Windows version 3.0 and later.)
PDERR_INITFAILURE	The PrintDlg function failed during initialization, and there is no more specific extended error code to describe the failure. This is the generic default error code for the function.
PDERR_LOADDRVFAILURE	The PrintDlg function failed to load the device driver for the specified printer.

PDERR_NODEFAULTPRN	A default printer does not exist.
PDERR_NODEVICES	No printer drivers were found.
PDERR_PARSEFAILURE	The PrintDlg function failed to parse the strings in the [devices] section of the WIN.INI file.
PDERR_PRINTERNOTFOUND	The [devices] section of the WIN.INI file did not contain an entry for the requested printer.
PDERR_RETDEFFAILURE	The PD_RETURNDEFAULT flag was specified in the Flags member of the PRINTDLG structure, but the hDevMode or hDevNames member was not NULL.
PDERR_SETUPFAILURE	The PrintDlg function failed to load the required resources.

The following error codes can be returned for the ChooseFont function.

Value	Meaning
CFERR_MAXLESSTHANMIN	The size specified in the nSizeMax member of the CHOOSEFONT structure is less than the size specified in the nSizeMin member.
CFERR_NOFONTS	No fonts exist.

The following error codes can be returned for the GetOpenFileName and GetSaveFileName functions.

Value	Meaning
FNERR_BUFFERTOOSMALL	The buffer pointed to by the lpstrFile member of the OPENFILENAME structure is too small for the file name specified by the user. The first two bytes of the lpstrFile buffer contain an integer value specifying the size, in TCHARs, required to receive the full name.
FNERR_INVALIDFILENAME	A file name is invalid.
FNERR_SUBCLASSFAILURE	An attempt to subclass a list box failed because sufficient memory was not available.

The following error code can be returned for the FindText and ReplaceText functions.

Value	Meaning
FRERR_BUFFERLENGTHZERO	A member of the FINDREPLACE structure points to an invalid buffer.

Requirements

Windows NT/2000: Requires Windows NT 3.1 or later.

Windows 95/98: Requires Windows 95 or later.

3.4 FindText

The FindText function creates a system-defined modeless Find dialog box that lets the user specify a string to search for and options to use when searching for text in a document.

```
FindText: procedure
(
    var lpfr    :FINDREPLACE
);
    stdcall;
    returns( "eax" );
    external( "__imp__FindTextA@4" );
```

Parameters

lpfr

[in] Pointer to a FINDREPLACE structure that contains information used to initialize the dialog box. The dialog box uses this structure to send information about the user's input to your application. For more information, see the following Remarks section.

Return Values

If the function succeeds, the return value is the window handle to the dialog box. You can use the window handle to communicate with or to close the dialog box.

If the function fails, the return value is NULL. To get extended error information, call the CommDlgExtendedError function. CommDlgExtendedError may return one of the following error codes:

CDERR_FINDRESFAILURE

CDERR_MEMLOCKFAILURE

CDERR_INITIALIZATION

CDERR_NOHINSTANCE

CDERR_LOCKRESFAILURE

CDERR_NOHOOK

CDERR_LOADRESFAILURE

CDERR_NOTEMPLATE

CDERR_LOADSTRFAILURE

CDERR_STRUCTSIZE

CDERR_MEMALLOCFAILURE

FRERR_BUFFERLENGTHZERO

Remarks

The FindText function does not perform a search operation. Instead, the dialog box sends FINDMSGSTRING registered messages to the window procedure of the owner window of the dialog box. When you create the dialog box, the hwndOwner member of the FINDREPLACE structure is a handle to the owner window.

Before calling FindText, you must call the RegisterWindowMessage function to get the identifier for the FINDMSGSTRING message. The dialog box procedure uses this identifier to send messages when the user clicks the Find Next button, or when the dialog box is closing. The lParam parameter of the FINDMSGSTRING message contains a pointer to a FINDREPLACE structure. The Flags member of this structure indicates the event that caused the message. Other members of the structure indicate the user's input.

If you create a Find dialog box, you must also use the IsDialogMessage function in the main message loop of your application to ensure that the dialog box correctly processes keyboard input, such as the TAB and ESC keys. IsDialogMessage returns a value that indicates whether the Find dialog box processed the message.

You can provide an FRHookProc hook procedure for a Find dialog box. The hook procedure can process messages sent to the dialog box. To enable a hook procedure, set the FR_ENABLEHOOK flag in the Flags member of the FINDREPLACE structure and specify the address of the hook procedure in the lpfnHook member.

Requirements

Windows NT/2000: Requires Windows NT 3.1 or later.

Windows 95/98: Requires Windows 95 or later.

3.5 GetFileTitle

The GetFileTitle function retrieves the name of the specified file.

```
GetFileTitle: procedure
(
        lpszFile    :string;
    var lpszTitle   :var;
        cbBuf       :word
);
    stdcall;
    returns( "eax" );
    external( "__imp__GetFileTitleA@12" );
```

Parameters

lpszFile

[in] Pointer to the name and location of a file.

lpszTitle

[out] Pointer to a buffer that receives the name of the file.

cbBuf

[in] Specifies the length, in TCHARs, of the buffer pointed to by the lpszTitle parameter. For the ANSI version of the function, this is in bytes; for the Unicode version, this is in characters.

Return Values

If the function succeeds, the return value is zero.

If the file name is invalid, the return value is unknown. If there is an error, the return value is a negative number.

If the buffer pointed to by the lpszTitle parameter is too small, the return value is a positive integer that specifies the required buffer size, in TCHARs. The required buffer size includes the terminating null character.

Remarks

GetFileTitle should only be called with legal Win32 file names; using an illegal file name has an unknown result.To get the buffer size needed for the name of a file, call the function with lpszTitle set to NULL and cbBuf set to zero. The function will return the required size.

GetFileTitle returns the string that the system would use to display the file name to the user. The display name includes an extension only if that is the user's preference for displaying file names. This means that the returned string may not accurately identify the file if it is used in calls to file system functions.

If the lpszTitle buffer is too small, GetFileTitle returns the size required to hold the display name. There is no guaranteed connection between the required size and the characters originally specified in the lpszFile buffer. In porting 16-bit Windows-based applications to Win32, you will need to update any code that relies on such behavior in previous versions of the system. The most common case is code that deliberately calls GetFileTitle with lpszTitle set to NULL and cbBuf set to zero, and then uses the return value as an index into the lpszFile string. This technique is no longer supported. You can usually achieve similar results (and superior performance) with run-time library functions such as strrchr, wcsrchr, and _mbsrchr.

Requirements

Windows NT/2000: Requires Windows NT 3.1 or later.

Windows 95/98: Requires Windows 95 or later.

3.6 GetOpenFileName

The GetOpenFileName function creates an Open dialog box that lets the user specify the drive, directory, and the name of a file or set of files to open.

```
BOOL GetOpenFileName(
  LPOPENFILENAME lpofn   // initialization data
);
```

Parameters

lpofn

[in/out] Pointer to an OPENFILENAME structure that contains information used to initialize the dialog box. When GetOpenFileName returns, this structure contains information about the user's file selection.

Return Values

If the user specifies a file name and clicks the OK button, the return value is nonzero. The buffer pointed to by the lpstrFile member of the OPENFILENAME structure contains the full path and file name specified by the user.

If the user cancels or closes the Open dialog box or an error occurs, the return value is zero. To get extended error information, call the CommDlgExtendedError function, which can return one of the following values:

```
CDERR_DIALOGFAILURE
CDERR_FINDRESFAILURE
CDERR_NOHINSTANCE
CDERR_INITIALIZATION
CDERR_NOHOOK
CDERR_LOCKRESFAILURE
CDERR_NOTEMPLATE
CDERR_LOADRESFAILURE
CDERR_STRUCTSIZE
CDERR_LOADSTRFAILURE
FNERR_BUFFERTOOSMALL
CDERR_MEMALLOCFAILURE
FNERR_INVALIDFILENAME
CDERR_MEMLOCKFAILURE
FNERR_SUBCLASSFAILURE
```

Remarks

Beginning with Windows 95 and Windows NT version 4.0, the default Open dialog box provides user-interface features that are similar to the Windows Explorer. You can provide an OFNHookProc hook procedure for an Explorer-style Open dialog box. To enable the hook procedure, set the OFN_EXPLORER and OFN_ENABLEHOOK flags in the Flags member of the OPENFILENAME structure and specify the address of the hook procedure in the lpfnHook member.

Windows 95/98 and Windows NT/Windows 2000 continue to support the old-style Open dialog box for applications that want to maintain a user-interface consistent with the Windows 3.1 or Windows NT 3.51 user-interface. To display the old-style Open dialog box, enable an OFNHookProcOldStyle hook procedure and ensure that the OFN_EXPLORER flag is not set.

To display a dialog box that allows the user to select a directory instead of a file, call the SHBrowseForFolder function.

With Windows 2000, the OPENFILENAME structure has increased to include some additional members. However, this causes problems for applications on previous operation systems. To use the current header files for applications on Windows 95/98 and Windows NT 4.0, either use the #define "/D_WIN32_WINNT=0x0400" or

use OPENFILENAME_SIZE_VERSION_400 for the lStructSize member of OPENFILENAME.

Requirements

Windows NT/2000: Requires Windows NT 3.1 or later.

Windows 95/98: Requires Windows 95 or later.

3.7 GetSaveFileName

The GetSaveFileName function creates a Save dialog box that lets the user specify the drive, directory, and name of a file to save.

```
GetSaveFileName: procedure
(
    var lpofn   :OPENFILENAME
);
    stdcall;
    returns( "eax" );
    external( "__imp__GetSaveFileNameA@4" );
```

Parameters

lpofn

[in/out] Pointer to an OPENFILENAME structure that contains information used to initialize the dialog box. When GetSaveFileName returns, this structure contains information about the user's file selection.

Return Values

If the user specifies a file name and clicks the OK button, the return value is nonzero. The buffer pointed to by the lpstrFile member of the OPENFILENAME structure contains the full path and file name specified by the user.

If the user cancels or closes the Save dialog box or an error occurs, the return value is zero. To get extended error information, call the CommDlgExtendedError function, which can return one of the following values:

CDERR_DIALOGFAILURE

CDERR_FINDRESFAILURE

CDERR_NOHINSTANCE

CDERR_INITIALIZATION

CDERR_NOHOOK

CDERR_LOCKRESFAILURE

CDERR_NOTEMPLATE

CDERR_LOADRESFAILURE

CDERR_STRUCTSIZE

CDERR_LOADSTRFAILURE

FNERR_BUFFERTOOSMALL

CDERR_MEMALLOCFAILURE

FNERR_INVALIDFILENAME

CDERR_MEMLOCKFAILURE

FNERR_SUBCLASSFAILURE

Remarks

By default, Windows 95 and Windows NT version 4.0 display a new version of the Save dialog box that provides user-interface features that are similar to the Windows Explorer. You can provide an OFNHookProc hook procedure for an Explorer-style Save dialog box. To enable the hook procedure, set the OFN_EXPLORER and OFN_ENABLEHOOK flags in the Flags member of the OPENFILENAME structure and specify the address of the hook procedure in the lpfnHook member.

Windows 95 and Windows NT 4.0 continue to support the old-style Save dialog box for applications that want to maintain a user-interface consistent with the Windows 3.1 or Windows NT 3.51 user-interface. To display the old-style Save dialog box, enable an OFNHookProcOldStyle hook procedure and ensure that the OFN_EXPLORER flag is not set.

With Windows 2000, the OPENFILENAME structure has increased to include some additional members. However, this causes problems for applications on previous operation systems. To use the current header files for applications on Windows 95/98 and Windows NT 4.0, either use the #define "/D_WIN32_WINNT=0x0400" or use OPENFILENAME_SIZE_VERSION_400 for the lStructSize member of OPENFILENAME.

Requirements

Windows NT/2000: Requires Windows NT 3.1 or later.

Windows 95/98: Requires Windows 95 or later.

3.8 PageSetupDlg

The PageSetupDlg function creates a Page Setup dialog box that enables the user to specify the attributes of a printed page. These attributes include the paper size and source, the page orientation (portrait or landscape), and the width of the page margins.

```
_PageSetupDlg: procedure
(
    var lppsd   :PAGESETUPDLG
);
    stdcall;
    returns( "eax" );
    external( "__imp__PageSetupDlgA@4" );
```

Parameters

lppsd

[in/out] Pointer to a PAGESETUPDLG structure that contains information used to initialize the dialog box. The structure receives information about the user's selections when the function returns.

Return Values

If the user clicks the OK button, the return value is nonzero. The members of the PAGESETUPDLG structure pointed to by the lppsd parameter indicate the user's selections.

If the user cancels or closes the Page Setup dialog box or an error occurs, the return value is zero. To get extended error information, use the CommDlgExtendedError function

Requirements

Windows NT/2000: Requires Windows NT 3.51 or later.

Windows 95/98: Requires Windows 95 or later.

3.9 PrintDlg

The PrintDlg function displays a Print dialog box. The Print dialog box enables the user to specify the properties of a particular print job.

Windows 2000: The PrintDlg function has been superseded by the PrintDlgEx function. PrintDlgEx displays a Print property sheet, which has a General page containing controls similar to the Print dialog box.

```
_PrintDlg: procedure
(
    var lppd    :PRINTDLG
);
    stdcall;
    returns( "eax" );
    external( "__imp__PrintDlgA@4" );
```

Parameters

lppd

[in/out] Pointer to a PRINTDLG structure that contains information used to initialize the dialog box. When PrintDlg returns, this structure contains information about the user's selections.

Return Values

If the user clicks the OK button, the return value is nonzero. The members of the PRINTDLG structure pointed to by the lppd parameter indicate the user's selections.

If the user canceled or closed the Print or Printer Setup dialog box or an error occurred, the return value is zero. To get extended error information, use the CommDlgExtendedError function. If the user canceled or closed the dialog box, CommDlgExtendedError returns zero; otherwise, it returns one of the following values.

Value	Value (continued)
CDERR_FINDRESFAILURE	PDERR_CREATEICFAILURE
CDERR_INITIALIZATION	PDERR_DEFAULTDIFFERENT
CDERR_LOADRESFAILURE	PDERR_DNDMMISMATCH
CDERR_LOADSTRFAILURE	PDERR_GETDEVMODEFAIL
CDERR_LOCKRESFAILURE	PDERR_INITFAILURE
CDERR_MEMALLOCFAILURE	PDERR_LOADDRVFAILURE
CDERR_MEMLOCKFAILURE	PDERR_NODEFAULTPRN
CDERR_NOHINSTANCE	PDERR_NODEVICES
CDERR_NOHOOK	PDERR_PARSEFAILURE
CDERR_NOTEMPLATE	PDERR_PRINTERNOTFOUND
CDERR_STRUCTSIZE	PDERR_RETDEFFAILURE

Remarks

If the hook procedure (pointed to by the lpfnPrintHook or lpfnSetupHook member of the PRINTDLG structure) processes the WM_CTLCOLORDLG message, the hook procedure must return a handle for the brush that should be used to paint the control background.

The PrintDlg function can also display a Print Setup dialog box. However, new applications should not use PrintDlg for this purpose. The Print Setup dialog box has been superseded by the Page Setup dialog box created by the PageSetupDlg function.

Requirements

Windows NT/2000: Requires Windows NT 3.1 or later.

Windows 95/98: Requires Windows 95 or later.

3.10 ReplaceText

The ReplaceText function creates a system-defined modeless dialog box that lets the user specify a string to search for and a replacement string, as well as options to control the find and replace operations.

```
ReplaceText: procedure
(
    var lpfr    :FINDREPLACE
);
    stdcall;
    returns( "eax" );
    external( "__imp__ReplaceTextA@4" );
```

Parameters

lpfr

[in/out] Pointer to a FINDREPLACE structure that contains information used to initialize the dialog box. The dialog box uses this structure to send information about the user's input to your application. For more information, see the following Remarks section.

Return Values

If the function succeeds, the return value is the window handle to the dialog box. You can use the window handle to communicate with the dialog box or close it.

If the function fails, the return value is NULL. To get extended error information, call the CommDlgExtendedError function, which can return one of the following error codes.

CDERR_FINDRESFAILURE

CDERR_MEMLOCKFAILURE

CDERR_INITIALIZATION

CDERR_NOHINSTANCE

CDERR_LOADRESFAILURE

CDERR_NOHOOK

CDERR_LOADSTRFAILURE

CDERR_NOTEMPLATE

CDERR_LOCKRESFAILURE

CDERR_STRUCTSIZE

CDERR_MEMALLOCFAILURE

FRERR_BUFFERLENGTHZERO

Remarks

The ReplaceText function does not perform a text replacement operation. Instead, the dialog box sends FINDMSGSTRING registered messages to the window procedure of the owner window of the dialog box. When you create the dialog box, the hwndOwner member of the FINDREPLACE structure is a handle to the owner window.

Before calling ReplaceText, you must call the RegisterWindowMessage function to get the identifier for the FINDMSGSTRING message. The dialog box procedure uses this identifier to send messages when the user clicks the Find Next, Replace, or Replace All buttons, or when the dialog box is closing. The lParam parameter of a FINDMSGSTRING message contains a pointer to the FINDREPLACE structure. The Flags member of this structure indicates the event that caused the message. Other members of the structure indicate the user's input.

If you create a Replace dialog box, you must also use the IsDialogMessage function in the main message loop of your application to ensure that the dialog box correctly processes keyboard input, such as the TAB and ESC keys. The IsDialogMessage function returns a value that indicates whether the Replace dialog box processed the message.

You can provide an FRHookProc hook procedure for a Replace dialog box. The hook procedure can process messages sent to the dialog box. To enable a hook procedure, set the FR_ENABLEHOOK flag in the Flags member of the FINDREPLACE structure and specify the address of the hook procedure in the lpfnHook member.

Requirements

Windows NT/2000: Requires Windows NT 3.1 or later.

Windows 95/98: Requires Windows 95 or later.