Object Reference

Microworks Custom Control Object Library
Version 1.03
for Borland Pascal 7.0 and Turbo Pascal for Windows 1.5

Overview

The MWCC Object Library provides you with a comprehensive set of 3-dimensional windows, dialogs and custom controls to use in your application development.

Using the library is easy as using Object Windows itself. Just derive your application objects from any of the objects in the MObjects unit and distribute your application with MWCC.DLL..

You wont need to use BWCC.DLL or CTL3D.DLL.. With the MWCC object library you can easily give your application's interface an enhanced Borland look, an enhanced Microsoft 3D look or use the library's own custom 3D style.

There are two distinct object classes in the library, MWCC objects and SFX objects. Except for TMDIWindow and TScroller there is a corresponding MWCC object for every Object Windows interface object.. The SFX objects are a custom set of windows and dialogs that resemble the Microsoft 3D look.

SeeAlso

Your Obligation
Changes in Version 1.03

Changes in Version 1.03

There have been several changes and some additions to this version but overall the library's usage remains the same. The source code has been refined, removing some unnecessary code and fixing a few minor problems, and there are three new objects - a common font dialog box, a mutiple selection list box and a cbs_Simple style 3D ComboBox.

A version of the object library for Borland C/C++ will be included in the next major release.

Changes

File List

MWCC03.ZIP contains the following files.

File	Description
Main.zip	Object units and library related files
Examples.zip	Basic window and dialog source code
HotKey1.zip	Task Hotkey sample application
HotKey2.zip	System Hotkey sample application
MDITool.zip	New toolbar in a MDI sample application
Message.zip	Complete Source code for non OWL Message Boxes
Rtl.zip	Run time library for the Object Units
WinMenu.zip	New 3D menu sample application
WinTask.zip	Graphical Windows task list sample application

Your Obligation

You are free to use the MWCC Object Library in your application's development and distribute your program with the file MWCC.DLL provided your program is distributed as shareware.

The MWCC Object Library may not be used in the development of any commercial software program in any way without prior negotiation of a commercial licence.

I hope to continue releasing the MWCC Object Library as freeware and will provide updates as it becomes necessary . If you would like to see something added, improved or changed then contact me.

Contact

Jeff Franks
Microworks.
33 Bayview St.
Waverley 2024
Australia.
Compuserve 100026,1134

Designing Your Interface

With everyone now jumping into Windows programming so many programmers make the same mistake. They give their program the same tired old Windows interface. Just imagine if every Windows program looked like Word for Windows. As things become more competitive many will soon realise that the way their program's interface looks and feels is just as important to the end user as what it does.

The MWCC Object library is a complete interface development library that's flexible, easy to use and looks great.

To help you use the MWCC object library what follows is a loose collection of tips and things you should know.

Resource Workshop.
The MWCC custom controls use ordinary Windows control objects. When placing the controls in your dialog templates, in order to align them you need to make allowance for the 3D border. List boxes, Static controls and Group boxes draw their border on the inside so the size of the Windows control will be the size of the MWCC 3D control. All other controls (edit controls, combo boxes etc) draw their 3D border on the outside so that these MWCC controls will appear 2 pixels bigger in all directions.

The MS Sans Serif 8 font is the best font size to use in your dialog box. because its clear and gives the best result. If you use another font the 3D controls may not appear correctly.

The SFX Frame and SFX style frame are based on the thick frame and not the moal frame. Always make sure you use a thick frame.

MWCC Button Objects.
A bitmapped button needs to have two positions in a dialog box, one for the VGASYS.FON system font and one for the 8514SYS.FON system font.

When you use a bitmapped button in a dialog box its size doesn't change between screen resolutions but the dialog box size does. For example, an MS Sans Serif 8 dialog box set up on a system that uses the VGASYS.FON system font will appear as big as if you had set up an MS Sans Serif 10 dialog box when displayed on a system using the 8514SYS.FON system font. You need to take this into account when writing your source code. The easiest way is to check if GetSystemMetrics(sm_CYSize) = 26. If it does the system font is 8514SYS.FON and you can move the buttons into an appropriate position. When you set up your dialog box in Resource Workshop you can simulate the different system fonts by setting it up at a font size of 8 and then changing to a font size 10. You can then calculate how far to move the buttons.

Any objects in this this library that use TMWCCBmpButton (eg. the common file open dialogs and the message boxes) shift the buttons for you.

Fonts
The default font is MS Sans Serif 8. MWCC custom controls give you a choice of a bold or regular font when used in a window. In a dialog box custom controls use the dialog box font.

Set up your dialog boxes up using the MS Sans Serif 8 font, it looks the best.
and gives the best results.

Static controls and Group boxes.
Always place static controls and group boxes in order first before any controls that appear in

front of them.

List Boxes.
Always check the Scroll Bar Always check box in the list box style dialog box in Resource Workshop.

Combo Boxes.
Always set the cbs_NoIntegralHeight attribute for a cbs_Simple style combo box.

This is done for you if the combo box is constructed using Init. When you construct a combo box with InitResource you will have to uncheck the *Integral Height* check box in Resource Workshop.

Common File Dialog Boxes.
You wont need to derive objects from TMWCCFileNameDlg or TSFXFileNameDlg as it has been done for you in the FileDlg unit. You can use the objects in this unit, TMWCCFileDlg and TSFXFileDlg in your application. Just customise the file dialog functions to suit your needs and add your version of the FileDlg unit to the uses clause.

You can re-arrange the dialog templates in MCommDlg.res to suit. If you do you might need to move the Ok and Cancel buttons. Use the button handles OkBtn^.HWindow and CancelBtn^.HWindow to do this.

If you want to add a bitmapped help button or some other special button don't use the OpenFileName flags. Add the button the same way you would to any other dialog box, by adding a WMDrawItem method.

Don't use the ws_ClipChildren style on 3D objects. It cuts off the 3D border.

MWCC Objects

Each MWCC object is directly descended from its Object Windows name sake and inherits all of its fields and methods. As well, each object declares some new fields and methods to handle its 3D painting and display. You should avoid overriding these declared methods.

Object Types

- TMWCCWindow
- TMWCCDlgWindow
- TMWCCDialog
- TMWCCFileNameDlg
- TMWCCFontDlg
- TMWCCListBox
- TMWCCComboBox
- TMWCCEdit
- TMWCCCheckBox
- TMWCCRadioButton
- TMWCCScrollBar
- TMWCCGroupBox
- TMWCCStatic
- TMWCCButton
- TMWCCBmpButton

SeeAlso

- MWCCMsgBox Function
- SFX Objects

SFX Objects

Each SFX object is directly descended from its Object Windows name sake and inherits all of its fields and methods with two exceptions. You can't use a regular menu bar or TScroller scroll bars with an SFX object but you can use pop up menus and TScrollbar scroll bars.

Unlike the Microsoft 3D look which just slaps a bit of paint over a dialog modal frame the SFX objects paint their own frame. A thick frame was used to overcome the title bar bitmap display problems associated the modal frame. How the frame is drawn around the caption bar depends on the thickness of the frame. To make the frame look effective and overcome the problem of variable frame thickness it is drawn in 3 ways, one for <= 3 pixels, one for 4 pixels and one for >= 5 pixels. The frame looks its best at a thickness of 5 pixels.

To ensure that your SFX object displays properly avoid overriding the declared methods.

Object Types

TSFXWindow
TSFXDlgWindow
TSFXDialog
TSFXFileNameDlg
TSFXFontDlg
TSFXListbox

SeeAlso

SFXMsgBox Function
MWCC Objects

TMWCCWindow

TMWCCWindow is a 3D window object directly descended from TWindow and inherits all of its fields and methods. Its main advantage is its flexible 3D interface. By default TMWCCWindow appears as a generic window with a light gray raised client area but you can easily add an SFX style frame, fix the window's size or give its background the chiselled steel look or a glazed appearance.

By default TMWCCWindow's Attr field is set to ws_PopupWindow or ws_Caption or ws_Thickframe or ws_Maximizebox or ws_Minimizebox or ws_Visible and not ws_VScroll and not ws_HScroll. You can change its Attr field to suit your needs. It you use an SFX style frame you will at least need to use ws_ThickFrame. Don't use ws_Ex_DlgModalFrame with an SFX style frame, it doesn't look good.

Fields

- IsSizeable
- SFXFrame

Methods

- Init
- Done (Never Override)
- GetClassName
- GetWindowClass
- SetUpWindow (Never Override)
- WMPaint
- WMCtlColor
- WMGetMinMaxInfo
- Miscellaneous Methods

SeeAlso

- MWCC Objects

TMWCCDlgWindow

TMWCCDlgWindow is a 3D dialog window object directly descended from TDlgWindow and inherits all of its fields and methods. Its main advantage is its flexible 3D interface. By default TMWCCDlgWindow appears as a generic window with a light gray raised client area but you can easily add an SFX style frame, fix the window's size or give its background the chiselled steel look or a glazed appearance.

When using an SFX style frame your resource template should use a thick frame and not a modal frame.

Fields

- IsSizeable
- SFXFrame

Methods

- Init
- Done (Never Override)
- GetClassName
- SetUpWindow (Never Override)
- WMPaint
- WMCltColor
- WMGetMinMaxInfo
- Miscellaneous Methods

SeeAlso

- MWCC Objects

TMWCCDialog

TMWCCDialog is a 3D dialog box object directly descended from TDialog and inherits all of its fields and methods. Its main advantage is its flexible 3D interface. By default TMWCCDialog appears as a generic dialog with a light gray raised client area but you can easily give its background the chiselled steel look or a glazed appearance.

TMWCCDialog doesn't support the use of an SFX style frame.

Methods

- Init
- Done (Never Override)
- GetClassName
- WMCltColor
- WMPaint

SeeAlso

- MWCC Objects

TMWCCFileNameDlg

TMWCCFileNameDlg is a 3D common file open dialog box object indirectly descended from TDialog and inherits all of its fields and methods. Its main advantage is its flexible 3D interface. By default TMWCCFileNameDlg appears as a generic dialog with a light gray raised client area but you can easily give its background the chiselled steel look or a glazed appearance.

TMWCCFileNameDlg doesn't support the use of an SFX style frame. Only the methods you need to access have been listed below.

TMWCCFileDlg in the FileDlg unit is descended from TMWCCFileNamedlg. You can use customise TMWCCFileDlg and use it to access all the necessary methods. Include your version of TMWCCFileDlg in your uses statement. The MDITool sample application shows you how to use TMWCCFileDlg in your application.

Fields

- OkBtn
- CancelBtn

Methods

- Init
- Done (Never Override)
- SetUpWindow (Never Override)
- DefSpec
- DefExt
- DefSpecPos
- OpenFlags
- CanClose
- DlgTitle WMCltColor
- WMPaint
- WMDrawItem

SeeAlso

- MWCC Objects

TMWCCFontDlg

TMWCCFontDlg is a 3D common font dialog box that is indirectly descended from TDialog. It has several specialized painting and information handling routines and child objects. You can use TMWCCFontDlg directly in your source code as shown in the MDITool sample application. The only time you would ever need to derive a new object from TMWCCFontDlg is when you want to add something, like an extra button. In such a case never override any of the inherited methods.

To find out all about TMWCCFontDlg and its methods have a look at MCommDlg.pas in RTL.zip

Ancestry

TMWCCFontDlg **>>** TChooseFontDlg **>>** TDialog

Child Objects

TFontGroupBox
TFontStatic
TFontComboBox
TFontCheckBox

Fields

OkBtn
CancelBtn

Methods

Init
Done
GetClassName
WMPaint

SeeAlso

MWCC Objects

TMWCCListBox

TMWCCListBox is a 3D list box object that is directly descended from TListbox and inherits all of its fields and methods. To properly draw the 3D frame the list box's scroll bar(s) must be permanently displayed. This is done for you in the Init constructor by setting the Attr field to include lbs_DisableNoScroll. When you use InitResource always check the Scroll Bar Always check box in the List Box Style dialog box in Resource Workshop.

Methods

Init
InitResource
Done (Never Override)
SetUpWindow (Never Override);
WMPaint

SeeAlso

MWCC Objects

TMWCCComboBox

TMWCCComboBox is a 3D combo box object that is directly descended from TCombobox and inherits all of its fields and methods. Only the cbs_DropDown and cbs_DropDownList styles are supported.

Methods

Init
InitResource
Done (Never Override)
SetUpWindow (Never Override);
WMPaint

SeeAlso

MWCC Objects

TMWCCEdit

TMWCCEdit is a 3D edit control object that is directly descended from TEdit and inherits all of its fields and methods. It supports both single line and multiple line edit controls.

Methods

Init
InitResource
Done (Never Override)
SetUpWindow (Never Override);
WMPaint

SeeAlso

MWCC Objects

TMWCCCheckBox

TMWCCCheckBox is a 3D check box object that is directly descended from TCheckBox and inherits all of its fields and methods.

Methods

- Init
- InitResource
- Done (Never Override)
- SetUpWindow (Never Override);
- WMPaint
- WMEnable
- BMSetCheck
- BMSetState

SeeAlso

- MWCC Objects

TMWCCRadioButton

TMWCCRadioButton is a 3D radio button object that is directly descended from TRadioButton and inherits all of its fields and methods.

Methods

- Init
- InitResource
- Done (Never Override)
- SetUpWindow (Never Override);
- WMPaint
- WMEnable
- BMSetCheck
- BMSetState

SeeAlso

- MWCC Objects

TMWCCScrollBar

TMWCCScrollBar is a 3D scroll bar object that is directly descended from TScrollBar and inherits all of its fields and methods.

Methods

- Init
- InitResource
- Done (Never Override)
- WMPaint

SeeAlso

- MWCC Objects

TMWCCGroupBox

TMWCCGroupBox is a 3D group box object that is directly descended from TGroupBox and inherits all of its fields and methods.

Creating the 3D group box wasn't as simple as drawing a 3D border. There was the title position, the empty middle and the need for a true group box to contend with. My solution was to create a new title positioned inside the 3D border (not on it) and fill in the group box with two light gray static objects, one for the caption and one for the background (painting the middle caused repainting problems).

Methods

- Init
- InitResource
- Done (Never Override)
- SetUpWindow (Never Override);
- WMPaint
- WMCtlColor
- WMSize
- WMEnable

SeeAlso

- MWCC Objects

TMWCCStatic

TMWCCStatic is a 3D static object that is directly descended from TStatic and inherits all of its fields and methods.

TMWCCStatic can be a recessed, raised or flat text static control. The MWCC and SFX objects use the WMCtlColor message to set the static control's colour to light gray. Recessed and raised static controls can be used to create 3D effects in your windows and dialogs. Only the flat static control supports the use of text.

To create humps and dips use a narrow static control (eg 5,6, or 7 pixels).

When placing a recessed or raised static in a resource template set the static control type to Black Frame.

Methods

Init
InitResource
Done (Never Override)
SetUpWindow (Never Override);
WMPaint

SeeAlso

MWCC Objects

TMWCCButton

TMWCCButton is an ownerdraw button object that is directly descended from TButton and inherits all of its fields and methods.

TMWCCButton is a special button object that can be used to create a Text button or a single-bitmapped button. It doesn't support the IsDefault field so you can't identify a default button. You will find this button object useful when creating special effects such as the menu bar in the Winmenu sample application.

The button display (text or bitmap) is automatically centred when the button is resized. The button has a special disabled view which grays text and finely hatches bitmaps.

To use TMWCCButton in your application you must include a WMDrawItem method. (See Ownerdraw.pas)

Fields

ResHdl

Methods

Init
Done (Never Override)
SetUpWindow (Never Override);
DrawItem

SeeAlso

MWCC Objects

TMWCCBmpButton

TMWCCBmpButton is an ownerdraw enhanced Borland style bitmapped button object that is directly descended from TButton and inherits all of its fields and methods. It's 74 by 54 pixels in size, it takes 3 bitmaps (up, down, and focussed) and it uses the Borland numbering system (+1000, +3000 and +5000). It does not support EGA graphics so there are no +2000, +4000 and +6000 bitmaps.

The 21 standard buttons are numbered form 1 to 21 and are stored in MWCC.DLL. They are,

1	Ok	8	Help	15	Open
2	Cancel	9	About	16	Save
3	Abort	10	Exit	17	Run
4	Retry	11	Browse	18	Font
5	Ignore	12	Icon	19	Cut
6	Yes	13	Option	20	Copy
7	No	14	Setup	21	Paste

You can easily add your own buttons to your application by including its 3 bitmaps in your resource file. If you only include 2 bitmaps (a 1000+ number and a 3000+ number) the button will not have a focussed view.

Your ID's must be between 100 and 999. 0 to 99 are reserved.

To use TMWCCBmpButton in your application you must include a WMDrawItem method. (See Ownerdraw.pas)

Fields

ResHdl

Methods

Init
Done (Never Override)
DrawItem
WMSetFocus

SeeAlso

MWCC Objects

TSFXWindow

TSFXWindow is a special 3D window object directly descended from TWindow. You can't use a regular menu bar or TScroller scroll bars with TSFXWindow but you can use popup menus and TScrollbar scroll bars.

Fields

- IsSizeable

Methods

- Init
- Done (Never Override)
- GetClassName
- GetWindowClass
- SetUpWindow (Never Override)
- WMPaint
- WMNCPaint
- WMCltColor
- WMNCCalcSize
- WMGetMinMaxInfo
- WMActivate
- WMNCActivate
- WMActivateApp

SeeAlso

- SFX Objects

TSFXDlgWindow

TSFXDlgWindow is a special 3D dialog window object directly descended from TDlgWindow. You can't use a regular menu bar or TScroller scroll bars with TSFXDlgWindow but you can use popup menus and TScrollbar scroll bars.

You should give your resource template a thick frame not a modal frame.

Fields

- IsSizeable

Methods

- Init
- Done (Never Override)
- GetClassName
- SetUpWindow (Never Override)
- WMCltColor
- WMPaint
- WMNCPaint
- WMNCCalcSize
- WMGetMinMaxInfo
- WMActivate
- WMNCActivate
- WMActivateApp

SeeAlso

- SFX Objects

TSFXDialog

TSFXDialog is a special 3D dialog object directly descended from TDialog. You can't use a regular menu bar or TScroller scroll bars with TSFXDialog but you can use popup menus and TScrollbar scroll bars.

You should give your resource template a thick frame not a modal frame.

Methods

- Init
- Done (Never Override)
- GetClassName
- SetUpWindow (Never Override)
- WMCltColor
- WMPaint
- WMNCCalcSize
- WMNCPaint
- WMGetMinMaxInfo
- WMActivate
- WMNCActivate
- WMActivateApp

SeeAlso

SFX Objects

TSFXFileNameDlg

TSFXFileNameDlg is a 3D common file open dialog box object indirectly descended from TDialog and inherits all of its fields and methods.

TSFXFileDlg in the FileDlg unit is descended from TSFXFileNamedlg. You can use customise TSFXFileDlg and use it to access all the necessary methods. Include your version of TSFXFileDlg in your uses statement. The MDITool sample application shows you how to use TSFXFileDlg in your application.

Fields

- OkBtn
- CancelBtn

Methods

- Init
- Done (Never Override)
- SetUpWindow (Never Override)
- DefSpec
- DefExt
- DefSpecPos
- OpenFlags
- CanClose
- DlgTitle
- WMCltColor
- WMDrawItem
- WMPaint
- WMNCCalcSize
- WMNCPaint
- WMGetMinMaxInfo
- WMActivate
- WMNCActivate
- WMActivateApp

SeeAlso

- SFX Objects

TSFXFontDlg

TSFXFontDlg is a 3D common font dialog box that is indirectly descended from TDialog. It has several specialized painting and information handling routines and child objects. You can use TSFXFontDlg directly in your source code as shown in the MDITool sample application. The only time you would ever need to derive a new object from TSFXFontDlg is when you want to add something, like an extra button. In such a case never override any of the inherited methods.

To find out all about TSFXFontDlg and its methods have a look at MCommDlg.pas in RTL.zip

Ancestry

TSFXFontDlg **>>** TChooseFontDlg **>>** TDialog

Child Objects

TFontGroupBox
TFontStatic
TFontComboBox
TFontCheckBox

Fields

OkBtn
CancelBtn

Methods

Init
Done
GetClassName
WMPaint

SeeAlso

SFX Objects

TSFXListBox

TSFXListBox is a mutilple selection list box that behaves as a single selection list box when you use the left mouse button and a multiple selection list box when you use the right mouse mouse button. It also allows up and down highlight scrolling of list box items without taking your finger off the right mouse button. You Should not override any of the declared methods.

Methods

Init
WMRButtonDown
WMRButtonUp
WMLButtonDown
WMMouseMove

SeeAlso

SFX Objects

MWCCMsgBox (Function)

function MWCCMsgBox (WndParent : HWnd ; ATxt , ACaption : PChar ; ATextType : Word ; ABmp : PChar) : Integer;

The **MWCCMsgBox** function creates, displays and operates an MWCC style message-box window. The message box contains an application defined message and title, plus any combination of predefined icons and push buttons specified in ATextType.

MWCCMsgBox operates identically to the Windows API MessageBox function and takes the same set of parameters.

There is only one difference,

1. MWCCMsgBox adds one field after ATextType, ABmp.

Like all the other MWCC windows and dialogs ABmp is the name of the bitmap to use as the background brush. ABmp can be,

'BWCC'	the Borland Chiselled Steel look
'MWCC'	the Microworks glazed look
NIL	Light Gray

SFXMsgBox (Function)

function SFXMsgBox (WndParent : HWnd ; ATxt , ACaption : PChar ; ATextType : Word) : Integer;

The **SFXMsgBox** function creates, displays and operates an SFX style message-box window. The message box contains an application defined message and title, plus any combination of predefined icons and push buttons specified in ATextType.

SFXMsgBox operates identically to the Windows API MessageBox function and takes the same set of parameters.

CenterOverClient (Procedure)

Procedure CenterOverClient (ParentWnd , Wnd : HWnd) ;

The CenterOverClient procedure can be used to centre a window over the client area of another window.

Parameters
ParentWnd identifies the window whose client area HWnd is to be centred over.
HWnd identifies the window that is to be centred over the client area of ParentWnd.

CenterOverWindow (Procedure)

Procedure CenterOverWindow (ParentWnd , Wnd : HWnd) ;

The CenterOverWindow procedure can be used to centre one window over another.

Parameters
ParentWnd identifies the window to centre HWnd over.
HWnd identifies the window that is to be centred over ParentWnd.

CenterOverScreen (Procedure)

Procedure CenterOverClient (Wnd : HWnd) ;

The CenterOverScreen procedure can be used to centre a window over the display screen.

Parameters
HWnd identifies the window that is to be centred over the display screen..

Draw3DBorder (Procedure)

Procedure Draw3DBorder (Wnd : HWnd ; X , Y , W , H : Integer ; Shade : Word) ;

Draw3DBorder can be used to draw either a raised or recessed 3D border. The MWCC controls use this procedure to draw their 3D borders.

Parameters

Wnd	Identifies the window that uses the border.
X	X coordinate of upper left corner of area to border
Y	Y coordinate of upper left corner of area to border
W	Width of area to border
H	Height of area to border
Shade	Type of border

Shade
Shade is the type of border to draw and can be one of the following values.

ctl_Recessed	(51)	Draws a recessed border
ctl_Raised (52)		Draws a raised border

Comments
The border is three lines thick, two lines for the border and one for its black margin. The border is drawn outwards starting at the supplied coordinates.
For example coordinates of X , Y , W , H will draw a border at X-2 , Y-2 , W+2 and H+2.

Draw3DFrame (Procedure)

Procedure Draw3DFrame (Wnd : HWnd) ;

Draw3DFrame draws the SFX style frame around TMWCCWindow and TMWCCDlgWindow. It gets declared in several object methods to ensure proper redrawing of the frame through all states of activation and inactivation. You will need to declare this procedure in any method that overrides the declared method, otherwise you wont need to use it.

Parameters
Wnd identifies the window that is using the SFX style frame.

DrawSFXFrame (Procedure)

Procedure DrawSFXFrame (Wnd : HWnd) ;

DrawSFXFrame draws the SFX frame around SFX windows and dialogs. It gets declared in several object methods to ensure proper redrawing of the frame through all states of activation and inactivation. You will need to declare this procedure in any method that overrides the declared method, otherwise you wont need to use it.

Parameters
Wnd identifies the window that is using the SFX frame.

TMWCCWindow.IsSizeable (Field)

Syntax

IsSizeable : Boolean; (read/write)

Description

When IsSizeable is True the window's thick frame can be resized. When it's False it can't. By default IsSizeable is True. You will find this most useful when your window uses an SFX style frame.

See Also

TMWCCWindow
MWCC Objects

TMWCCWindow.SFXFrame (Field)

Syntax

SFXFrame : Boolean; (read/write)

Description

When SFXFrame is True the window dislplays an SFX style frame. When it's False it displays a generic frame. By default SFXFrame is False.

See Also

TMWCCWindow
MWCC Objects

TMWCCWindow.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AName , ABmp : PChar) ;

Description

Init adds one extra field to the TWindow Init method, ABmp, which is the name of the bitmap pattern used to create the background brush.

ABmp can be,

'BWCC'	the Borland Chislled Steel look
'MWCC'	the Microworks glazed look
NIL	Light Gray

The default Attr field is set to ws_PopupWindow or ws_Caption or ws_ThickFrame or ws_MinimizeBox or ws_MaximizeBox or ws_Visible and not ws_VScroll and not ws_HScroll;

IsSizeable is set to True.
SFXFrame is set to False.

See Also

TMWCCWindow
MWCC Objects

TMWCCWindow.GetClassName (Method)

Syntax

Usual Syntax

Description

(Override : Sometimes) Declares the class name MWCCWindow

See Also

TMWCCWindow
MWCC Objects

TMWCCWindow.GetWindowClass (Method)

Syntax

Usual Syntax

Description

The declared GetWindowClass method only sets one field, the AWndClass.HbrBackground field. If ABmp specifies either the BWCC or MWCC then HbrBackground is set to a brush for that pattern, otherwise it's set to GetStockObject(LtGray_Brush). You should never override the AWndClass.HbrBackground field.

You can override all the other AWndclass fields.

See Also

TMWCCWindow
MWCC Objects

TMWCCWindow.WMPaint (Method)

Syntax

Usual Syntax

Description

The wm_Paint method uses the procedure Draw3DBorder to draw a raised border around the client area.

You should only override this method if you want to change the dimensions of the border as was done in the WinMenu sample application. In this case you would need to add the Draw3DBorder procedure to your wm_Paint method.

See Also

WMPaint.pas
TMWCCWindow
MWCC Objects

TMWCCWindow.WMCtlColor (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the colour of CtlColor_Btn, and CtlColor_Static to light gray.

See Also

TMWCCWindow
MWCC Objects

TMWCCWindow.WMGetMinMaxInfo (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the thick frame size. If IsSizeable is True the thick frame can be resized. If it is False it can't.

See Also

TMWCCWindow
MWCC Objects

Miscellaneous Methods

The following methods are used by TMWCCWindow when you use an SFX style frame.

WMActivate
WMActivateApp
WMNCPaint
WMNCActivate

These methods are used to ensure that the SFX style frame is re-drawn properly through all states of window activation and inactivation. In general you should not override these methods if you use an SFX stlye frame.

All these messages do the same thing. They use the procedure Draw3DFrame and call the inherited TWindow method. If there isn't one TWindow's DefWndProc is called. If you have to override one of these methods just add the line Draw3DFrame(HWindow) to your method.

See Also

TMWCCWindow
MWCC Objects

TMWCCDlgWindow.IsSizeable (Field)

Syntax

IsSizeable : Boolean; (read/write)

Description

When IsSizeable is True the window's thick frame can be resized. When it's False it cannot. By default IsSizeable is True. You will find this of most use when your window uses an SFX style frame.

See Also

TMWCCDlgWindow
MWCC Objects

TMWCCDlgWindow.SFXFrame (Field)

Syntax

SFXFrame : Boolean; (read/write)

Description

When SFXFrame is True the window dislplays an SFX style frame. When it's False it displays a generic frame. By default SFXFrame is False.

See Also

TMWCCDlgWindow
MWCC Objects

TMWCCDlgWindow.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AName , ABmp : PChar) ;

Description

Init adds one extra field to the TDlgWindow Init method, ABmp, which is the name of the bitmap pattern used to create the background brush.

ABmp can be,

'BWCC'	the Borland Chiselled Steel look
'MWCC'	the Microworks glazed look
NIL	Light Gray

IsSizeable is set to True.
SFXFrame is set to False.

See Also

TMWCCDlgWindow
MWCC Objects

TMWCCDlgWindow.GetClassName (Method)

Syntax

Usual Syntax

Description

(Override : Sometimes) Declares the class name MWCCDlgWindow

See Also

TMWCCDlgWindow
MWCC Objects

TMWCCDlgWindow.WMCtlColor (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the colour of CtlColor_Btn, and CtlColor_Static to light gray and CtlColor_Dlg to the background bitmap brush created using ABmp. If ABmp is NIL CtlColor_Dlg is set to GetStockObject(LtGray_Brush).

See Also

TMWCCDlgWindow
MWCC Objects

TMWCCDlgWindow.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) The WMPaint method uses the procedure Draw3DBorder to draw a raised border around the client area.

You should only override the WMPaint method if you want to change the dimensions of the border as was done in the WinMenu sample application. In this case you would need to add the Draw3DBorder procedure to your WMPaint method.

See Also

WMPaint.pas
TMWCCDlgWindow
MWCC Objects

TMWCCDlgWindow.WMGetMinMaxInfo (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the thick frame size. If IsSizeable is True the thick frame can be resized. If it is False it can't.

See Also

TMWCCDlgWindow
MWCC Objects

Miscellaneous Methods

The following methods are used by TMWCCDlgWindow when you use an SFX style frame.

WMActivate
WMActivateApp
WMNCPaint
WMNCActivate

These methods are used to ensure that the SFX style frame is re-drawn properly through all states of window activation and inactivation. In general you should not override these methodsif you use an SFX stlye frame

All these messages do the same thing. They use the procedure Draw3DFrame and call the inherited TDlgWindow method. If there isn't one DefWndProc is called. If you have to override one of these methods just add the line Draw3DFrame(HWindow) to your method.

See Also

TMWCCDlgWindow
MWCC Objects

TMWCCDialog.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AName , ABmp : PChar) ;

Description

Init adds one extra field to the TDialog Init method, ABmp, which is the name of the bitmap pattern used to create the background brush.

ABmp can be,

'BWCC'	the Borland Chiselled Steel look
'MWCC'	the Microworks glazed look
NIL	Light Gray

See Also

TMWCCDialog
MWCC Objects

TMWCCDialog.GetClassName (Method)

Syntax

Usual Syntax

Description

(Override : Sometimes) Declares the class name MWCCDialog

See Also

TMWCCDialog
MWCC Objects

TMWCCDialog.WMCtlColor (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the colour of CtlColor_Btn, and CtlColor_Static to light gray and CtlColor_Dlg to the bitmap brush created using ABmp. If ABmp is NIL then CtlColor_Dlg is set to GetStockObject(LtGray_Brush).

See Also

TMWCCDialog
MWCC Objects

TMWCCDialog.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) The WMPaint method uses the procedure Draw3DBorder to draw a raised border around the client area.

You should only override the WMPaint method if you want to change the dimensions of the border as was done in the WinMenu sample application. In this case you would need to add the Draw3DBorder procedure to your WMPaint method.

See Also

WMPaint.pas
TMWCCDialog
MWCC Objects

TMWCCFileNameDlg.OkBtn (Field)

Syntax

OkBtn : PMWCCBmpButton;

Description

OkBtn is a PWindowsObject pointer to the Ok button in the file name dialog box. You can use this pointer to shift the Ok button if you rearrange the default template.

See Also

ShiftButton.Pas
TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.CancelBtn (Field)

Syntax

CancelBtn : PMWCCBmpButton;

Description

CancelBtn is a PWindowsObject pointer to the Cancel button in the file name dialog box. You can use this pointer to shift the Cancel button if you rearrange the default template.

See Also

ShiftButton.Pas
TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AName , ABmp : PChar ; IsOpen : Boolean) vb;

Description

Init adds two extra fields to the TDialog Init method, ABmp and IsOpen. ABmp is the name of the bitmap pattern used to create the background brush.

ABmp can be,

'BWCC'	the Borland Chiselled Steel look
'MWCC'	the Microworks glazed look
NIL	Light Gray

IsOpen specifies which common file dialog it is. If IsOpen is True it's a File Open dialog box. If it's False it's a Save As dialog box.

The default edit controls, combo boxes and list boxes are replaced by 3D MWCC controls.

The Ok (OkBtn) and Cancel (CancelBtn) buttons are initialised.

See Also

TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.WMCtlColor (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the colour of CtlColor_Btn, and CtlColor_Static to light gray and CtlColor_Dlg to the bitmap brush created using ABmp. If ABmp is NIL then CtlColor_Dlg is set to GetStockObject(LtGray_Brush).

See Also

TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) The WMPaint method uses the procedure Draw3DBorder to draw a raised border around the client area.

You should only override the WMPaint method if you want to change the dimensions of the border as was done in the WinMenu sample application. In this case you would need to add the Draw3DBorder procedure to your WMPaint method

See Also

WMPaint.pas
TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.WMDrawItem (Method)

Syntax

Usual Syntax

Description

(Override : Never) Draws the ownerdraw Ok and Cancel TMWCCBmpButton objects.

See Also

Ownerdraw.pas
TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.DefSpec (Method)

Syntax

function DefSpec : PChar;

Description

(Override : Always) Describes the file specifications to list in the List Files of Type combo box.

See Also

DefSpec.pas
TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.DefExt (Method)

Syntax

function DefExt : PChar;

Description

(Override : Always) This specifies the default extension of the files to display in the file dialog box when it first appears. It is one of the extensions listed in the DefSpec function.

See Also

DefExt.pas
TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.DefSpecPos (Method)

Syntax

function DefSpecPos : Byte;

Description

(Override : Always) This specifies the position of the default extension in the list of extensions given in the DefSpec function (eg 1, 2, 3 or 4 etc);

See Also

DefSpecPos.pas
TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.OpenFlags (Method)

Syntax

function OpenFlags : LongInt;

Description

(Override : Sometimes) This specifies the OpenFileName flags to use. The default flags are,

for the File Open dialog box:

ofn_PathMustExist or ofn_HideReadOnly

>

and for the File Save As dialog box:

ofn_PathMustExist or ofn_HideReadOnly or ofn_NoReadOnlyReturn;

>

See Also

OpenFlags.pas
TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.CanClose (Method)

Syntax

function CanClose : Boolean;

Description

(Override : Sometimes) This calls the default CanClose method.

See Also

CanClose.pas
TMWCCFileNameDlg
MWCC Objects

TMWCCFileNameDlg.DlgTitle (Method)

Syntax

function DlgTitle : PChar;

Description

(Override : Sometimes) This lets you change the default dialog titles which are File Open for a file open dialog box and File Save As for a file save as dialog box.

See Also

DlgTitle.pas
TMWCCFileNameDlg
MWCC Objects

TMWCCFontDlg.OkBtn (Field)

Syntax

OkBtn : PMWCCBmpButton;

Description

OkBtn is a PWindowsObject pointer to the Ok button in the common font dialog box. You can use this pointer to shift the Ok button if you rearrange the default template.

See Also

ShiftButton.Pas
TMWCCFontDlg
MWCC Objects

TMWCCFontDlg.CancelBtn (Field)

Syntax

CancelBtn : PMWCCBmpButton;

Description

CancelBtn is a PWindowsObject pointer to the Cancel button in the file name dialog box. You can use this pointer to shift the Cancel button if you rearrange the default template.

See Also

ShiftButton.Pas
TMWCCFontDlg
MWCC Objects

TMWCCListBox.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AnId : Integer ; X , Y , W , H : Integer ; IsBold : Boolean) ;

Description

Init adds only one field to the TListbox's Init method, IsBold. IsBold refers to the default list box font. If IsBold is True the default font is bold. If it's False the default font is regular.

The default list box font is set to MS Sans Serif 8.

The list box's Attr field is set to Attr.Style or lbs_DisableNoScroll. Do not override these attributes. When adding attributes always call Attr.Style.

See Also

TMWCCListBox
MWCC Objects

TMWCCListBox.InitResource (Method)

Syntax

Constructor InitResource (AParent : PWindowsObject ; AnId : Integer) ;

Description

Adds no new fields to the TListBox InitResource method.

Use a normal Windows list box control in your resource template.

To ensure proper 3D painting always make sure you check the Scroll Bar Always check box in the List Box Style dialog box in Resource Workshop.

See Also

TMWCCListBox
MWCC Objects

TMWCCListBox.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMpaint checks to see if the list box has its vertical scroll bar and/or horizontal scroll bar attributes set. If either have been set allowance for them is made when drawing the 3D border. If no scroll bar is found the scroll bar that appears when the list box is full will be outside the 3D frame.

See Also

TMWCCListBox
MWCC Objects

TMWCCComboBox.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AnId : Integer ; X , Y , W , H : Integer ; AStyle , ATextLen : Word ; IsBold : Boolean) ;

Description

Init adds only one field to the TComboBox Init method, IsBold. IsBold refers to the default combo box font. If IsBold is True the default font is bold. If it's False the default font is regular.

The default combo box font is set to MS Sans Serif 8.

All three combo box styles are supported (cbs_Simple, cbs_DropDown and cbs_DropDownList).

See Also

TMWCCComboBox
MWCC Objects

TMWCCComboBox.InitResource (Method)

Syntax

Constructor InitResource (AParent : PWindowsObject ; AnId : Integer ; ATextlen : Word) ;

Description

Adds no new fields to the TComboBox InitResource method.

Use a normal Windows combo box control in your resource template.

All three combo box styles are supported (cbs_Simple, cbs_DropDown and cbs_DropDownList). Because of the special painting required to draw the cbs_Simple and cbs_DropDown styles you should always use the MS San Serif 8 (vga/supervga) or 10(extended vga) dialog font size.

See Also

TMWCCComboBox
MWCC Objects

TMWCCComboBox.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMpaint uses thwe DrawComboBox paint procedure in MWCC.DLL. DrawComboBox checks to see which attribute style is set, cbs_Simple, cbs_DropDown or cbs_DropDownList, and draws an appropriate 3D border.

See Also

TMWCCComboBox
MWCC Objects

TMWCCEdit.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AnId : Integer ; ATitle : PChar ; X , Y , W , H , ATextLen : Integer ; Multiline , IsBold : Boolean) ;

Description

Init adds only one field to the TEdit Init method, IsBold. IsBold refers to the default edit control font. If IsBold is True the default font is bold. If it's False the default font is regular.

The default edit control font is set to MS Sans Serif 8.

See Also

TMWCCEdit
MWCC Objects

TMWCCEdit.InitResource (Method)

Syntax

Constructor InitResource (AParent : PWindowsObject ; AnId : Integer ; ATextlen : Word) ;

Description

Adds no new fields to the TEdit InitResource method.

Use a normal Windows edit control in your resource template.

See Also

TMWCCEdit
MWCC Objects

TMWCCEdit.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMpaint checks to see if the es_Multiline attribute is set and if it is it makes allowances for the scroll bar(s) when drawing the 3D border.

See Also

TMWCCEdit
MWCC Objects

TMWCCCheckBox.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AnId : Integer ; ATitle : PChar ; X , Y , W , H : Integer ; AGroup : PGroupBox ; IsBold : Boolean) ;

Description

Init adds only one field to the TCheckBox Init method, IsBold. IsBold refers to the default check box font. If IsBold is True the default font is bold. If it's False the default font is regular.

The default check box font is set to MS Sans Serif 8.

See Also

WMCtlColor.pas
TMWCCCheckBox
MWCC Objects

TMWCCCheckbox.InitResource (Method)

Syntax

Constructor InitResource (AParent : PWindowsObject ; AnId : Integer) ;

Description

Adds no new fields to the TCheckbox InitResource method.

Use a normal Windows check box control in your resource template.

See Also

WMCtlColor.pas
TMWCCCheckBox
MWCC Objects

TMWCCCheckBox.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMPaint checks to see if a large or small system font is being used and draws an appropriately sized 3D border around the check box.

See Also

TMWCCCheckBox
MWCC Objects

TMWCCCheckBox.WMEnable (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMEnable is one of four methods used to ensure proper redrawing of the 3D check box through all stages of activation and inactivation. It sends a WMPaint message without calling DefWndProc.

See Also

TMWCCCheckBox
MWCC Objects

TMWCCCheckBox.BMSetCheck (Method)

Syntax

Usual Syntax

Description

(Override : Never) BMSetCheck is used to ensure proper redrawing of the 3D check box through all stages of activation and inactivation by sending the check box a WMPaint message. Never override the inherited BMSetCheck Methed.

See Also

TMWCCCheckBox
MWCC Objects

TMWCCCheckBox.BMSetState (Method)

Syntax

Usual Syntax

Description

(Override : Never) BMSetState is used to ensure proper redrawing of the 3D check box through all stages of activation and inactivation by sending the check box a WMPaint message. Never override the inherited BMSetState Methed.

See Also

TMWCCCheckBox
MWCC Objects

TMWCCRadioButton.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AnId : Integer ; ATitle : PChar ; X , Y , W , H : Integer ; AGroup : PGroupBox ; IsBold: Boolean) ;

Description

Init adds only one field to the TRadioButton Init method, IsBold. IsBold refers to the default radio button font. If IsBold is True the default font is bold. If it's False the default font is regular.

The default radio button font is set to MS Sans Serif 8.

See Also

WMCtlColor.pas
TMWCCRadioButton
MWCC Objects

TMWCCRadioButton.InitResource (Method)

Syntax

Constructor InitResource (AParent : PWindowsObject ; AnId : Integer) ;

Description

Adds no new fields to the TRadioButton InitResource method.

Use a normal Windows radio button control in your resource template.

See Also

WMCtlColor.pas
TMWCCRadioButton
MWCC Objects

TMWCCRadioButton.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMPaint checks to see if a large or small system font is being used and draws an appropriately sized 3D border around the radio button.

See Also

TMWCCRadioButton
MWCC Objects

TMWCCRadioButton.WMEnable (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMEnable is one of four methods used to ensure proper redrawing of the 3D radio button through all stages of activation and inactivation. It sends a WMPaint message without calling DefWndProc.

See Also

TMWCCRadioButton
MWCC Objects

TMWCCRadioButton.BMSetCheck (Method)

Syntax

Usual Syntax

Description

(Override : Never) BMSetCheck is used to ensure proper redrawing of the 3D radio button through all stages of activation and inactivation by sending the radio button a WMPaint message. Never override the inherited BMSetCheck Methed.

See Also

TMWCCRadioButton
MWCC Objects

TMWCCRadioButton.BMSetState (Method)

Syntax

Usual Syntax

Description

(Override : Never) BMSetState is used to ensure proper redrawing of the 3D radio button through all stages of activation and inactivation by sending the radio button a WMPaint message. Never override the inherited BMSetState Methed.

See Also

TMWCCRadioButton
MWCC Objects

TMWCCScrolBar.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AnId : Integer ; X , Y , W , H : Integer ; IsHScrollBar : Boolean) ;

Description

Init adds no new fields to the TScrollBar Init method.

See Also

TMWCCScrollBar
MWCC Objects

TMWCCScrolBar.InitResource (Method)

Syntax

Constructor InitResource (AParent : PWindowsObject ; AnId : Integer) ;

Description

Adds no new fields to the TScrollBar InitResource method.

Use a normal Windows scroll bar control in your resource template.

See Also

TMWCCScrollBar
MWCC Objects

TMWCCScrolBar.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMPaint calls the inherited WMPaint method and draws a 3D border around the scroll bar.

See Also

TMWCCScrollBar
MWCC Objects

TMWCCGroupBox.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AnId : Integer ; AText : PChar ; X , Y , W , H : Integer ; IsBold : Boolean) ;

Description

Init adds only one field to the TGroupBox Init method, IsBold. IsBold refers to the default group box title font. If IsBold is True the default font is bold. If it's False the default font is regular.

The default group box title font is set to MS Sans Serif 8. Unlike the other controls you can't override this default font.

Two TMWCCStatic objects are initialised, one for the caption and one for the background.

Always initialise the group box in the constructor before the controls it is to group and never use a group box initialised using Init with controls initialised using InitResource.

See Also

TMWCCGroupBox
MWCC Objects

TMWCCGroupBox.InitResource (Method)

Syntax

Constructor InitResource (AParent : PWindowsObject ; AnId : Integer ; AText : PChar);

Description

InitResource adds one new field to the TGroupBox InitResource method, AText. AText is the title of the group box. You must use AText for the title not the resource template.

Two TMWCCStatic objects are initialised, one for the caption and one for the background.

Always place the group box in the resource template before the controls it is to group and don't use resource controls with a group box initialised using Init.

See Also

TMWCCGroupBox
MWCC Objects

TMWCCGroupBox.WMCtlColor (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMCtlColor sets the CtlColor_Static field to light gray.

See Also

TMWCCGroupBox
MWCC Objects

TMWCCGroupBox.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMPaint draws the 3D border around the group box..

See Also

TMWCCGroupBox
MWCC Objects

TMWCCGroupBox.WMSize (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMSize re-sizes the two static objects if the group box is resized.

See Also

TMWCCGroupBox
MWCC Objects

TMWCCGroupBox.WMEnable (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMEnable disables/enables the group box title static object to match the disable/enable state of the group box and sends the group box a WMPaint message. To prevent the default Windows group box from being re-drawn DefWndProc is not called.

See Also

TMWCCGroupBox
MWCC Objects

TMWCCStatic.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AnId : Integer ; ATitle : PChar ; X , Y , W , H : Integer ; ATextLen , AShade : Word ; IsBold : Boolean) ;

Description

Init adds two new fields to the TStatic Init method, AShade and IsBold. AShade can be one of the following.

ctl_Static		creates a flat static control for text input
(50)		
ctl_Recessed	(51)	creates a recessed static control (no text)
ctl_Raised		creates a raised static control (no text)
(52)		

IsBold refers to the default static control font. If IsBold is True the default font is bold. If it's False the default font is regular.

The default font is set to MS Sans Serif 8.

Always initialise the static control in the constructor before any controls that are to appear in front of it.

See Also

WMCtlColor.pas
TMWCCStatic
MWCC Objects

TMWCCStatic.InitResource (Method)

Syntax

Constructor InitResource (AParent : PWindowsObject ; AnId : Integer ; ATextLen , AShade : Word) ;

Description

InitResource adds one new field to the TStatic InitResource method, AShade. AShade can be one of the following,

ctl_Static (50)		creates a flat static control for text input
ctl_Recessed	(51)	creates a recessed static control (no text)
ctl_Raised (52)		creates a raised static control (no text)

Always place the static control in the resource template before any controls that are to appear in front of it.

See Also

WMCtlColor.pas
TMWCCStatic
MWCC Objects

TMWCCStatic.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) If ctl_Recessed or ctl_Raised is specified WMPaint draws an appropriate 3D border around the static control.

See Also

TMWCCStatic
MWCC Objects

TMWCCButton.ResHdl (Field)

Syntax

ResHdl : THandle; (read/write)

Description

ResHdl holds the handle of the module that stores the button bitmaps. By default it's HInstance so that the bitmaps get loaded from the EXE file. If you want to override HInstance and load the bitmaps from your own DLL set Reshdl to the Handle returned by Loadlibrary.

See Also

TMWCCButton
MWCC Objects

TMWCCButton.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AnID : Integer ; AText : PChar ; X , Y , W , H : Integer ; ABmp : PChar ; AStyle : Word) ;

Description

Init adds two new fields, ABmp and AStyle, and removes one field, IsDefault, from the TButton Init method.

ABmp is the title of a single Bitmap to display on the button. The button can display text or a bitmap. If any text is specified the bitmap is ignored. To display a bitmap you should set AText to Nil.

AStyle specifies whether to draw the button flush or raised. It can be one of the following values.

ctl_Flush	(53)	draws the button with a flush border
null (0)		draws a regular raised button

IsDefault has been removed to prevent thick black lines etc. being drawn around the buttons.

If AText is specified the default font is set to MS Sans Serif 8 (regular).

By default ResHdl is set to HInstance;

See Also

TMWCCButton
MWCC Objects

TMWCCDrawItem (Method)

Syntax

Procedure DrawItem (var Msg: TMessage);

Description

(Override : Never) DrawItem uses an internal MWCC.DLL procedure call to draw the ownerdraw button(s).

See Also

TMWCCButton
MWCC Objects

TMWCCBmpButton.ResHdl(Field)

Syntax

ResHdl : THandle; (read/write)

Description

ResHdl holds the handle of the module that stores the button bitmaps. By default it is HInstance so that the bitmaps get loaded from the EXE file. If you want to override HInstance and load the bitmaps from your own DLL set Reshdl to the Handle returned by Loadlibrary.

See Also

TMWCCBmpButton
MWCC Objects

TMWCCBmpButton.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AnID , X , Y : Integer ; IsDefault : Boolean , iBmp : Integer ; AStyle : Word) ;

Description

Init adds two new fields, iBmp and AStyle, to the TButton Init method.

iBmp is the ID number of the bitmaps minus 1000, 3000 and 5000 (eg for 1201, 3201 and 5201 iBmp = 201). Usually this will also be the ID number of the button but it doesn't have to be. To use one of the standard buttons in MWCC.DLL enter its ID number (1 to 21) in the iBmp field. Init adds 1000, 3000, and 5000 to iBmp and loads the corresponding bitmaps. If no 5000+ bitmap is found the 1000+ bitmap is used in its place. iBmp must be between 100 and 999. 0 to 99 have been reserved. If iBmp is less than 100 Init loads the bitmaps from MWCC.DLL. By default, if iBmp is between 100 and 999 Init loads the bitmaps from the EXE file. If you wan to override this and load the bitmaps from your own DLL set ResHdl to the handle return by loadlibrary.

ctl_Flush	(53)	draws the button with a flush border
null (0)		draws a regular raised button

By default ResHdl is set to HInstance.

See Also

TMWCCBmpButton
MWCC Objects

TMWCCBmpButton.DrawItem (Method)

Syntax

Procedure DrawItem (var Msg: TMessage);

Description

(Override : Never) DrawItem uses an internal MWCC.DLL procedure call to draw the ownerdraw button(s).

See Also

TMWCCBmpButton
MWCC Objects

TMWCCBmpButton.WMSetFocus (Method)

Syntax

Usual Syntax

Description

Override : Never) WMSetFocus is used to redraw the button (showing the focussed view) when the button receives the input focus.

See Also

TMWCCBmpButton
MWCC Objects

TSFXWindow.IsSizeable (Field)

Syntax

IsSizeable : Boolean; (read/write)

Description

When IsSizeable is True the window's thick frame can be resized. When it's False it cannot. By default IsSizeable is True.

See Also

TSFXWindow
SFX Objects

TSFXWindow.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AName : PChar) ;

Description

Init adds no new fields to the TWindow Init method.

The Window's Attr field is set to ws_PopupWindow or ws_Caption or ws_ThickFrame or ws_MinimizeBox or ws_MaximizeBox or ws_Visible and not ws_VScroll and not ws_HScroll;

You should always call Attr.Style and then add on your changes.

Just in case you accidentally override the Attr field ws_ThickFrame and not ws_VScroll and not ws_HScroll are reset in the SetUpWindow method. SetUpWindow also checks for a class menu and if found destroys it. This is necessary to ensure that the window displays properly.

IsSizeable is set to True;

See Also

TSFXWindow
SFX Objects

TSFXWindow.GetClassName (Method)

Syntax

Usual Syntax

Description

(Override : Sometimes) Declares the class name SFXWindow

See Also

TSFXWindow
SFX Objects

TSFXWindow.GetWindowClass (Method)

Syntax

Usual Syntax

Description

The declared GetWindowClass method sets only one field, AWndClass.HbrBackground, to GetStockObject(LtGray_Brush). You should never override the HbrBackground field.

See Also

TSFXWindow
SFX Objects

TSFXWindow.WMCtlColor (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the colour of CtlColor_Btn, and CtlColor_Static to light gray.

See Also

TSFXWindow
SFX Objects

TSFXWindow.WMNCCalcSize (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMNCCalcSize shrinks the client area of the window and passes the new measurements onto to DefWindowProc.

See Also

WMNCCalcSize.pas
TSFXWindow
SFX Objects

TSFXWindow.WMNCPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMNCPaint overrides the default ncpaint method by not calling DefWndProc. This prevents the Frame, Menu and Scroll Bars from being drawn. The new ncpaint method draws an SFX frame and Title bar but not the menu or scroll bars.

See Also

TSFXWindow
SFX Objects

TSFXWindow.WMGetMinMaxInfo (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the thick frame size. If IsSizeable is True the thick frame can be resized. If it is False it can't.

See Also

TSFXWindow
SFX Objects

TSFXWindow.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited TWindow WMPaint method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMPaint behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXWindow
SFX Objects

TSFXWindow.WMActivate (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited TWindow WMActivate method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMActivate behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXWindow
SFX Objects

TSFXWindow.WMNCActivate (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMNCActivate behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXWindow
SFX Objects

TSFXWindow.WMActivateApp (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMActivateApp behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXWindow
SFX Objects

TSFXDlgWindow.IsSizeable (Field)

Syntax

IsSizeable : Boolean; (read/write)

Description

When IsSizeable is True the window's thick frame can be resized. When it's False it cannot. By default IsSizeable is True.

See Also

TSFXDlgWindow
SFX Objects

TSFXDlgWindow.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AName : PChar) ;

Description

Init adds no new fields to the TDlgWindow Init method.

The window's attributes ws_ThickFrame and not ws_VScroll and not ws_HScroll are set in the SetUpWindow method. SetUpWindow also checks for a class menu and if found destroys it.. This is necessary to ensure that the window displays properly.

IsSizeable is set to True;

See Also

TSFXDlgWindow
SFX Objects

TSFXDlgWindow.GetClassName (Method)

Syntax

Usual Syntax

Description

(Override : Sometimes) Declares the class name SFXDlgWindow

See Also

TSFXDlgWindow
SFX Objects

TSFXDlgWindow.WMCtlColor (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the colour of ctlcolor_Dlg, CtlColor_Btn, and CtlColor_Static to light gray.

See Also

TSFXDlgWindow
SFX Objects

TSFXDlgWindow.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Seldom) WMPaint calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need to override the default WMPaint method make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXDlgWindow
SFX Objects

TSFXDlgWindow.WMNCCalcSize (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMNCCalcSize shrinks the client area of the window and passes the new measurements onto to DefWindowProc.

See Also

WMNCCalcSize.pas
TSFXDlgWindow
SFX Objects

TSFXDlgWindow.WMNCPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMNCPaint overrides the default ncpaint method by not calling DefWndProc. This prevents the Frame, Menu and Scroll Bars from being drawn. The new ncpaint method draws an SFX frame and Title bar but not the menu or scroll bars.

See Also

TSFXDlgWindow
SFX Objects

TSFXDlgWindow.WMGetMinMaxInfo (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the thick frame size. If IsSizeable is True the thick frame can be resized. If it is False it can't.

See Also

TSFXDlgWindow
SFX Objects

TSFXDlgWindow.WMActivate

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMActivate behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXDlgWindow
SFX Objects

TSFXDlgWindow.WMNCActivate (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMNCActivate behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXDlgWindow
SFX Objects

TSFXDlgWindow.WMActivateApp (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMActivateApp behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXDlgWindow
SFX Objects

TSFXDialog.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AName : PChar) ;

Description

Init adds no new fields to the TDialog Init method.

The dialog's attributes ws_ThickFrame and not ws_VScroll and not ws_HScroll are set in the SetUpWindow method. SetUpWindow also checks for a class menu and if found destroys it.. This is necessary to ensure that the dialog displays properly.

See Also

TSFXDialog
SFX Objects

TSFXDialog.GetClassName(Method)

Syntax

Usual Syntax

Description

(Override : Sometimes) Declares the class name SFXDialog

See Also

TSFXDialog
SFX Objects

TSFXDialog.WMCtlColor (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the colour of ctlcolor_Dlg, CtlColor_Btn, and CtlColor_Static to light gray.

See Also

TSFXDialog
SFX Objects

TSFXDialog.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Seldom) WMPaint calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need to override the default WMPaint method make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXDialog
SFX Objects

TSFXDialog.WMNCCalcSize (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMNCCalcSize shrinks the client area of the dialog and passes the new measurements onto to DefWindowProc.

See Also

WMNCCalcSize.pas
TSFXDialog
SFX Objects

TSFXDialog.WMNCPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMNCPaint overrides the default ncpaint method by not calling DefWndProc. This prevents the Frame, Menu and Scroll Bars from being drawn. The new ncpaint method draws an SFX frame and Title bar but not the menu or scroll bars.

See Also

TSFXDialog
SFX Objects

TSFXDialog.WMGetMinMaxInfo (Method)

Syntax

Usual Syntax

Description

(Override : Never) Prevents the thick frame from being resized.

See Also

TSFXDialog
SFX Objects

TSFXDialog.WMActivate (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMActivate behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXDialog
SFX Objects

TSFXDialog.WMNCActivate (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMNCActivate behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXDialog
SFX Objects

TSFXDialog.WMActivateApp (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMActivateApp behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXDialog
SFX Objects

TSFXFileNameDlg.OkBtn (Field)

Syntax

OkBtn : PMWCCBmpButton;

Description

OkBtn is a PWindowsObject pointer to the Ok button in the file name dialog box. You can use this pointer to shift the Ok button if you rearrange the default template.

See Also

ShiftButton.Pas
TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.CancelBtn (Field)

Syntax

CancelBtn : PMWCCBmpButton;

Description

CancelBtn is a PWindowsObject pointer to the Cancel button in the file name dialog box. You can use this pointer to shift the Cancel button if you rearrange the default template.

See Also

ShiftButton.Pas
TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.Init (Method)

Syntax

Constructor Init (AParent : PWindowsObject ; AName : PChar ; IsOpen : Boolean) ;

Description

Init adds one extra field to the TDialog Init method, IsOpen. IsOpen specifies which common file dialog to use. If IsOpen is True it's a File Open dialog box. If it's False it's a Save As dialog box.

The default edit controls, combo boxes and list boxes are replaced by 3D MWCC controls.

The Ok (OkBtn) and Cancel (CancelBtn) buttons are initialised.

See Also

TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.WMCtlColor (Method)

Syntax

Usual Syntax

Description

(Override : Never) Sets the colour of CtlColor_Dlg, CtlColor_Btn, and CtlColor_Static to light gray.

See Also

TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.WMPaint (Method)

Syntax

Usual Syntax

Description

(Override : Seldom) WMPaint calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need to override the default WMPaint method make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.WMDrawItem(Method)

Syntax

Usual Syntax

Description

(Override : Never) Draws the ownerdraw Ok and Cancel TMWCCBmpButton objects

See Also

Ownerdraw.pas
TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.DefSpec (Method)

Syntax

function DefSpec : PChar;

Description

(Override : Always) Describes the file specifications to list in the List Files of Type combo box.

See Also

DefSpec.pas
TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.DefExt (Method)

Syntax

function DefExt : PChar;

Description

(Override : Always) This specifies the default extension of the files to display in the file dialog box when it first appears. It is one of the extensions listed in the DefSpec function.

See Also

DefExt.pas
TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.DefSpecPos (Method)

Syntax

function DefSpecPos : Byte;

Description

(Override : Always) This specifies the position of the default extension in the list of extensions given in the DefSpec function (eg 1, 2, 3 or 4 etc);

See Also

DefSpecPos.pas
TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.OpenFlags (Method)

Syntax

function OpenFlags : LongInt;

Description

(Override : Sometimes) This specifies the OpenFileName flags to use. The default flagsare,

for the File Open dialog:

ofn_PathMustExist or ofn_HideReadOnly
>

and for the Save As dialog:

ofn_PathMustExist or ofn_HideReadOnly or ofn_NoReadOnlyReturn;
>

See Also

OpenFlags.pas
TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.CanClose (Method)

Syntax

function CanClose : Boolean;

Description

(Override : Sometimes) This calls the default CanClose method.

See Also

CanClose.pas
TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.DlgTitle (Method)

Syntax

function DlgTitle : PChar;

Description

(Override : Sometimes) This lets you change the default dialog titles which are File Open for a file open dialog box and File Save As for a file save as dialog box.

See Also

DlgTitle.pas
TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.WMNCCalcSize (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMNCCalcSize shrinks the client area of the dialog and passes the new measurements onto to DefWindowProc.

See Also

WMNCCalcSize.pas
TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.WMNCPaint (Method)

Syntax

Usual Syntax

Description

(Override : Never) WMNCPaint overrides the default ncpaint method by not calling DefWndProc. This prevents the Frame, Menu and Scroll Bars from being drawn. The new ncpaint method draws an SFX frame and Title bar but not the menu or scroll bars.

See Also

TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.WMGetMinMaxInfo (Method)

Syntax

Usual Syntax

Description

(Override : Never) Prevents the thick frame from being resized.

See Also

TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.WMActivate (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMActivate behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.WMNCActivate (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMNCActivate behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXFileNameDlg
SFX Objects

TSFXFileNameDlg.WMActivateApp (Method)

Syntax

Usual Syntax

Description

(Override : Rarely) Calls the inherited DefWndProc method and uses the procedure DrawSFXFrame. Should you ever need the override the default WMActivateApp behaviour make sure you add the line DrawSFXFrame(HWindow) to your method.

See Also

TSFXFileNameDlg
SFX Objects

TSFXFontDlg.OkBtn (Field)

Syntax

OkBtn : PMWCCBmpButton;

Description

OkBtn is a PWindowsObject pointer to the Ok button in the common font dialog box. You can use this pointer to shift the Ok button if you rearrange the default template.

See Also

ShiftButton.Pas
TSFXFontDlg
SFX Objects

TSFXFontDlg.CancelBtn (Field)

Syntax

CancelBtn : PMWCCBmpButton;

Description

CancelBtn is a PWindowsObject pointer to the Cancel button in the file name dialog box. You can use this pointer to shift the Cancel button if you rearrange the default template.

See Also

ShiftButton.Pas
TSFXFontDlg
SFX Objects

Main.zip

Main.zip contains all the library related files.

File	Description
FileDlg.pas	Souce code for filedlg unit (Common file dialogs)
FileDlg.tpu	FileDlg unit for TPW 1.5
FileDlg.tpw	FileDlg unit for BP 7.0
MWCC.dll	Re-distributable custom control dynamic link library
MWCC.hlp	Windows help file
MCommDlg.res	Common file dialog templates for object unit
MCommDlg.tpu	Common dialog object unit for TPW 1.5
MCommDlg.tpw	Common dialog object unit for BP 7.0
MMsgBox.tpu	Message box unit for TPW 1.5
MMsgBox.tpw	Message box unit for BP 7.0
MObjects.tpu	Object unit for TPW 1.5
MObjects.tpw	Object unit for BP 7.0
Viewer.exe	Object viewer to display all the custom control objects

WMPaint.pas

To override the inherited WMPaint method you must add the following lines of code to your method.

```
procedure TExample.WMPaint (var Msg: TMessage);
var
        PS    : TPaintStruct;
        W, H : Integer;
begin
        GetClientRect (HWindow, CRect);
        W := CRect.Right;
        H := CRect.Bottom;
        BeginPaint(HWindow, PS);
        Draw3DBorder(HWindow, 1, 1, W-2, H-2, ctl_Raised);{Default Values}
        /
        /
        /
        EndPaint(HWindow, PS);
end;
```

ShiftButton.pas

MoveWindow(OkBtn^.HWindow, X, Y, 74, 54, True);

or

SetWindowPos(OkBtn^.HWindow, 0, X, Y, 0, 0, swp_NoSize or swp_NoZOrder);

ShiftButton.pas

MoveWindow(CancelBtn^.HWindow, X, Y, 74, 54, True);

or

SetWindowPos(CancelBtn^.HWindow, 0, X, Y, 0, 0, swp_NoSize or swp_NoZOrder);

DefSpec.pas

```
function TExample.DefSpec: PChar;
begin
        DefSpec := 'All Files (*.*)'#0'*.*'#0'TextFiles (*.TXT)'#0'*.TXT'#0;{Add Specs here}
end;
```

DefExt.pas

```
function TExample.DefExt: PChar;
begin
        DefExt := 'TXT';
end;
```

DefSpecPos.pas

```
function TExample.DefSpecPos: Byte;
begin
        DefSpecPos := 2;
end;
```

OpenFlags.pas

The default flags.

```
function TExample.OpenFlags: LongInt;
begin
        OpenFlags := TMWCCFileNameDlg.OpenFlags;
end;
```

Changing the default flags.

```
function TExample.OpenFlags: LongInt;
begin
        OpenFlags := TMWCCFileNameDlg.OpenFlags and not ofn_HideReadOnly;
end;
```

CanClose.pas

```
function TExample.CanClose: Boolean;
begin
        CanClose := TMWCCFileNameDlg.CanClose;
end;
```

DlgTitle.pas

```
function TExample.DlgTitle :PChar;
begin
        DlgTitle := TMWCCFileNameDlg.DlgTitle;
end;
```

or

```
function TExample.DlgTitle :PChar;
begin
        DlgTitle := 'Program Browse';
end;
```

WMCtlColor.pas

The MWCC and SFX window and dialog objects use the WMCtlColor message to set the colour of the Check boxes, Radio buttons and Static controls to light gray. If you use any of these controls in your own window or dialog objects you should use WMCtlColor to set their colour to light gray. Use ctlcolor_Btn for check boxes and radio buttons and ctlcolor_Static for static controls.

```
procedure TExample.WMCtlColor (var Msg: TMessage);
begin
        case Msg.LParamHi of
                CtlColor_Static:
                begin
                        SetBkMode(Msg.wParam, Transparent);
                        Msg.Result := GetStockObject(LtGray_Brush);
                end
                else
                        inherited DefWndProc(Msg);
        end;
end;
```

or

```
procedure TExample.WMCtlColor (var Msg: TMessage);
begin
        case Msg.LParamHi of
                CtlColor_Btn, CtlColor_Static:
                begin
                        SetBkMode(Msg.WParam, Transparent);
                        Msg.Result := GetStockObject(LtGray_Brush);
                end
                else
                        inherited DefWndProc(Msg);
        end;
end;
```

OwnerDraw.pas

To add an ownerdraw TMWCCButton or TMWCCBmpButton object to your application you need to include something similar to the following lines of code. In this example 'OkBut' uses the Windows ID - id_Ok (1).

```
const
        id_But1 = 201;

type

        PExample = ^TExample;
        TExample = object(TWindow)
                But1 : PMWCCButton;
                OkBut : PMWCCBmpButton;
                constructor Init (AParent: PWindowsObject; AName: PChar);
                        procedure WMDrawItem (var Msg: TMessage);  virtual   wm_First +
wm_DrawItem;
                                        /
                                        /
                                        /
                 end;

constructor   TExample.Init (AParent:   PWindowsObject; AName: PChar);
begin
        TWindow.Init (AParent, AName);
        But1 := New(PMWCCButton, Init(@Self, id_But1, 'Push Button', X, Y, W, H, nil,
ctl_Flush
        OkBut := New(PMWCCBmpButton, Init(@Self, id_Ok, X, Y, False, 1,   ctl_Flush
        /
        /
        /
end;

procedure TExample.WMDrawItem (var Msg: TMessage);
begin
        with PDrawItemStruct(Msg.lParam)^ do
                 case CtlType of odt_Button:
                        case CtlID of
                        id_But1 : But1^.DrawItem(Msg);
                        id_Ok : OkBut^.DrawItem(Msg);
                end;
        end;
end;
```

WMNCCalcSize.pas

This example is only mentioned for interest sake. Most of the time you wont need to use it. However, the odd occasion might arise when you want to paint a special non-client area and need to shrink the client area of the window accordingly. Using this method you can easily shrink the client area in any direction. The SFX objects use this method to shrink the top by 1 pixel, not much but necessary to improve the redrawing of the window. To shrink the left, right or bottom sides just use the left, right or bottom field of TRect in the Inc method. The number 1 is the number of pixels to inc by. The most important part is the pass the altered wm_NCCalcSize message onto DefWindowProc.

```
procedure TExample.WMNCCalcSize (var Msg: TMessage);
type
        CRect = array[0..2] of TRect;
        PRect = ^CRect;
begin
        Inc(PRect(Msg.lParam)^[0].Top, 1);
        DefWindowProc(HWindow, wm_NCCalcSize, Msg.wParam, Msg.lParam);
end;
```

OpenFlags.pas

The default flags.

```
function TExample.OpenFlags: LongInt;
begin
        OpenFlags := TSFXFileNameDlg.OpenFlags;
end;
```

Changing the default flags.

```
function TExample.OpenFlags: LongInt;
begin
        OpenFlags := TSFXFileNameDlg.OpenFlags and not ofn_HideReadOnly;
end;
```

CanClose.pas

```
function TExample.CanClose: Boolean;
begin
        CanClose := TSFXFileNameDlg.CanClose;
end;
```

DlgTitle.pas

```
function TExample.DlgTitle :PChar;
begin
        DlgTitle := TSFXFileNameDlg.DlgTitle;
end;
```

or

```
function TExample.DlgTitle :PChar;
begin
        DlgTitle := 'Program Browse';
end;
```

MObjects Unit

The MObjects unit contains all the MWCC and SFX objects (excluding Common Dialogs).

There are two versions of this unit, **MObjects.tpu** for Turbo Pascal 1.5 and **MObjects.tpw** for Borland Pascal 7.0. This unit replaces the MWCC unit found in earlier versions.

MWCCMsgBox

MWCCMsgBox (WndParent : HWnd ; ATxt , ACaption : PChar ; ATextType : Word ; ABmp : PChar);

SFXMsgBox

SFXMsgBox (WndParent : HWnd ; ATxt , ACaption : PChar ; ATextType : Word);

MWCC.DLL

MWCC.DLL is the re-distributable file that you must package with your application. It contains all the custom drawing and painting functions used in the object units and the 21 standard TMWCCBmpButton bitmaps. MWCC.DLL must be kept in the Windows system subdirectory.

MWCC.DLL is compatible with TPW 1.5 and BP 7.0.

As it becomes necessary to add to or enhance MWCC.DLL updates will be made freely available. You should instruct your users to always use the latest version of MWCC.DLL.

TMWCCBmpButton

TMWCCBmpButton is an enhanced Borland style bitmapped button object. The button is 74 by 54 pixels in size, takes three bitmaps and uses the Borland numbering system. There are 21 standard bitmapped buttons numbered from 1 to 21.

Using an SFX Style Frame

If you use an SFXFrame but don't use a menu bar or declare ws_VScroll and/or ws_HScroll in the window's Attr field then the wm_NCPaint message (which draws the frame) is not passed onto DefWndProc. This prevents the generic frame from being drawn and you wont see its colour flash when the window is resized. If you use a menu bar and/or scroll bars the message gets passed onto DefWndProc so the menu bar and/or scroll bars can be drawn. This also means that the generic frame gets drawn and you will see its colour flash when the window is resized. The importance of all this has to do with your interface design and what you think looks good. I prefer not to see flashing colours.

Default Window Attributes

The default. attributes for TMWCCWindow and TSFXWindow are now

ws_PopupWindow or ws_Caption or ws_ThickFrame or ws_MinimizeBox or ws_MaximizeBox or ws_Visible and not ws_VScroll and not ws_HScroll;

Object Units

The MWCC object unit has been broken up into three smaller units, **MObjects**- the main object unit, **MMsgBox** - the message box unit and **MCommDlg** - the common dialog unit.

If you want to use the *MWCCMsgBox* or *SFXMsgBox* functions in your soucre code just add MMsgBox to the uses clause.

If you want to use an MWCC or SFX style common dialog add MCommDlg to the uses clause, customize the FileDlg unit (FileDlg.pas) to suit your application's needs and add it to the uses clause, and include the appropriate dialog box template in your resource file (from MCommDlg.res).

To update your existing source code ammend its uses clause replacing MWCC with **MObjects**. Add **MMsgBox** if you have used the MWCCMsgbox or SFXMsgBox functions and change the PWindowsObject pointer **@Self** to **HWindow** or **null** (0). If you have used either the File Open or File Save As common dialogs add **MCommDlg** to the uses clause.

SeeAlso

MObjects Unit
MMsgBox Unit
MCommDlg Unit
Changes in Version 1.03

Message Boxes

The MWCC and SFX message boxes have been rewritten without using Objects Windows to resolve a few working problems. The only noticeable difference is that the two message box functions, *MWCCMsgBox* and *SFXMsgBox*, now take a window handle as the first parameter (like the Windows API MessageBox function) and not a PWindowsObject pointer . This means that the message boxes can now be used as stand alone message boxes (eg within a constructor) by specifing a null (0) window handle.

The complete source code for the message boxes (including its code from MWCC.DLL) is included in Message.zip as an example of how to write a non-OWL Windows application in Borland Pascal.

SeeAlso

Changes in Version 1.03

MWCC.DLL

MWCC.DLL has been rewritten and includes the painting,drawing and keydown message box procedures and three new functions CenterOverClient, CenterOverWindow and CenterOverScreen. There are two new buttons, a new number 10 button (exit) and new number 18 button (font). The export indices have also changed.

SeeAlso

Changes in Version 1.03

Default Window Attributes

The default window attributes for TMWCCWindow and TSFXWindow have been changed to overcome a minor display problem with the caption bitmaps.

SeeAlso

Changes in Version 1.03

Font Dialog Box

One of the major improvements is the addition of an MWCC and SFX style Font dialog box. After a lot of research I found that if you wanted to paint a common font dialog box you had to include your own sample text paint method. As you'll see from the source code I wrote my own text paint routine and used the choosefont *lpCustdata* field to customize the sample text. The end result is an excellent font dialog box that really works. An example of how to use a font dialog box can be found in the updated MDITool sample application.

Static number 1093 in the dialog box template
If the flags cf_Both or cf_PrinterFonts are set, static control 1093 displays the text that tells you '*This font is a true type font ...*'.

In the font dialog box templates in MCommDlg.res I placed this static control inside the *Sample* group box. The Sample text paint method in MCommDlg looks for this static control. If it's found inside the group box the sample text is painted between the top of the static and the top of the group box. If it's found outside the group box the sample text is painted in the centre. If neither the cf_Both or cf_PrinterFonts flag are set you should place this static control outside the dialog box so that the sample text centred.

If you don't set the cf_effects flag you can place the *Color* combo box and the *Effects* group box outside the dialog box (don't delete them) and resize the *Sample* group box to fill the empty space.

SeeAlso

Changes in Version 1.03

TSFXListBox

The new TSFXListBox object is a refined version of the list box sample I wrote and uploaded to the Compuserve BPascal forum as *Multi.zip.*

TSFXListBox is a mutilple selection list box that behaves as a single selection list box when you use the left mouse button and a multiple selection list box when you use the right mouse mouse button. It also allows up and down highlight scrolling of list box items without taking your finger off the right mouse button.

SeeAlso

Changes in Version 1.03

TMWCCWindow

TMWCCWindow no longer has a wm_Size or wm_Move method and the Paint method has been replaced by a wm_Paint method.

SeeAlso

Changes in Version 1.03

TMWCCDlgWindow

TMWCCDlgWindow no longer has a wm_Size or wm_Move method.

SeeAlso

Changes in Version 1.03

TMWCCDialog

TMWCCDialog no longer has a SetupWindow method.

SeeAlso

Changes in Version 1.03

TMWCCComboBox

TMWCCComboBox now supports the cbs_Simple style. A sample of this new 3D style can be seen in the Font dialog box in the MDITool sample application.

The **cbs_simple** style must be used with the **cbs_NoIntegralHeight** attribute. The inherited SetUpWindow method sets the cbs_NoIntegralHeight attribute for cbs_Simple combo boxes initialized with Init. It has no effect on combo boxes initialized with InitResource so you will have to uncheck the *integral height* check box in Resource Workshop.

SeeAlso

Changes in Version 1.03

ResHdl (THandle)

A **ResHdl** field has been added to **TMWCCButton** and **TMWCCBmpButton**. It holds a handle to the module that stores the button bitmaps. By default it's HInstance so that your button bitmaps get loaded from the exe file. (For ID's < 100 ResHdl points to MWCC.dll.)

You can use ResHdl to overide the default *HInstance* and load button bitmaps from your own DLL.by setting ResHdl to the handle returned by Loadlibrary. ResHdl should be declared in your constructor after you call the inherited init method.

SeeAlso

Changes in Version 1.03

TSFXWindow

TSFXWindow no longer has a wm_Size method and now has a wm_Paint method.

SeeAlso

Changes in Version 1.03

TSFXDlgWindow

TSFXDlgWindow no longer has a wm_Size method.

SeeAlso

Changes in Version 1.03

MObjects Unit

The MObjects unit contains all the MWCC and SFX objects (excluding Common Dialogs).

There are two versions of this unit, **MObjects.tpu** for Turbo Pascal 1.5 and **MObjects.tpw** for Borland Pascal 7.0. This unit replaces the MWCC unit found in earlier versions.

SeeAlso

MMsgBox Unit
MCommDlg Unit
Changes in Version 1.03

MCommDlg Unit

The MCommDlg unit contains all the common dialog objects. So far only two classes are included - the ChooseFont dialog box and the File Open and File Save As dialog box. Each class has two styles, an MWCC style and an SFX style to match the other MWCC and SFX objects found in the MObjects unit. The MCommdlg unit comes with the file MCommDlg.res which contains four dialog box templates.

There are two versions of this unit, **MCommDlg.tpu** for Turbo Pascal 1.5 and **MCommDlg.tpw** for Borland Pascal 7.0. This unit is new to version 1.03.

SeeAlso

MObjects Unit
MMsgBox Unit
Changes in Version 1.03

MMsgBox Unit

The MMsgBox unit contains the MWCC and SFX message boxes. To use these message boxes in your application add **MMsgBox** to your source code's uses clause and use the MWCCMsgBox or SFXMsgBox function. Both functions work in exactly the same way as the Windows API MessageBox function and take the same set of parameters (MWCCMsgbox takes one more).

There are two versions of this unit, **MMsgBox.tpu** for Turbo Pascal 1.5 and **MMsgBox.tpw** for Borland Pascal 7.0. This unit is new to version 1.03

SeeAlso

MObjects Unit
MCommDlg Unit
Changes in Version 1.03

Error Messages

The MWCC.DLL error messages have been removed from all object constructors. It is now up to you to include your own error message if MWCC.dll is not found in the Windows system subdirectory (optional).

SeeAlso

Changes in Version 1.03