
ModernUI Documentation

Release 1.0

mr fearless

Jul 18, 2019

1	Introduction	3
1.1	Installation ModernUI	3
1.2	Setup ModernUI	4
2	ModernUI Library	5
2.1	ModernUI Base Functions	5
2.1.1	MUIGetIntProperty	5
2.1.2	MUISetIntProperty	6
2.1.3	MUIGetExtProperty	7
2.1.4	MUISetExtProperty	7
2.2	ModernUI DPI Functions	8
2.3	ModernUI Font Functions	8
2.3.1	MUIPointSizeToLogicalUnit	8
2.4	ModernUI GDI Functions	9
2.4.1	MUIGDIBlend	9
2.4.2	MUIGDIBlendBitmaps	9
2.4.3	MUIGDIDoubleBufferFinish	10
2.4.4	MUIGDIDoubleBufferStart	10
2.4.5	MUIGDIPaintFill	11
2.4.6	MUIGDIPaintFrame	11
2.4.7	MUIGDIRotateCenterBitmap	12
2.4.8	MUIGDIStretchBitmap	12
2.4.9	MUIGDIStretchImage	13
2.5	ModernUI GDIPlus Functions	13
2.5.1	MUIGDIPlusDoubleBufferFinish	13
2.5.2	MUIGDIPlusDoubleBufferStart	14
2.5.3	MUIGDIPlusFinish	15
2.5.4	MUIGDIPlusPaintFill	16
2.5.5	MUIGDIPlusPaintFillI	17
2.5.6	MUIGDIPlusPaintFrame	17
2.5.7	MUIGDIPlusPaintFrameI	18
2.5.8	MUIGDIPlusRectToGdipRect	19
2.5.9	MUIGDIPlusRotateCenterImage	20
2.5.10	MUIGDIPlusStart	20
2.5.11	MUILoadPngFromResource	21
2.6	ModernUI Image Functions	21

2.6.1	MUICreateBitmapFromMemory	21
2.6.2	MUICreateCursorFromMemory	22
2.6.3	MUICreateIconFromMemory	23
2.6.4	MUIGetImageSize	23
2.6.5	MUIGetImageSizeEx	24
2.6.6	MUILoadBitmapFromResource	24
2.6.7	MUILoadIconFromResource	25
2.6.8	MUILoadImageFromResource	25
2.7	ModernUI Memory Functions	25
2.7.1	MUIAllocMemProperties	25
2.7.2	MUIFreeMemProperties	26
2.7.3	MUIAllocStructureMemory	27
2.8	ModernUI Painting Functions	28
2.8.1	MUIGetParentBackgroundBitmap	28
2.8.2	MUIGetParentBackgroundColor	29
2.8.3	MUIPaintBackground	29
2.8.4	MUIPaintBackgroundImage	30
2.9	ModernUI Region Functions	31
2.9.1	MUILoadRegionFromResource	31
2.9.2	MUISetRegionFromResource	32
2.10	ModernUI Window Functions	33
2.10.1	MUIApplyToDialog	33
2.10.2	MUICenterWindow	33
2.10.3	MUIGetParentRelativeWindowRect	34
2.11	ModernUI Macros	34
2.11.1	MUI_RGBCOLOR Macro	34
2.11.2	MUI_ARGBCOLOR Macro	35
2.11.3	RGB Macro	35
2.11.4	AlphaRGB Macro	36
2.12	ModernUI Library Data Types	36
2.12.1	ModernUI Data Types	36
2.12.2	GDI+ Data Types	37
2.12.3	Common Data Types	37
2.12.4	Data Types Description	37
2.13	ModernUI Library Structures	39
2.13.1	GDIPRECT Members	39
3	ModernUI Controls	41
3.1	ModernUI_Button	41
3.1.1	ModernUI_Button Functions	41
3.1.2	ModernUI_Button Messages	48
3.1.3	ModernUI_Button Properties	49
3.2	ModernUI_CaptionBar	50
3.2.1	ModernUI_CaptionBar Functions	50
3.2.2	ModernUI_CaptionBar Messages	54
3.2.3	ModernUI_CaptionBar Properties	54
3.3	ModernUI_Checkbox	54
3.3.1	ModernUI_Checkbox Functions	54
3.3.2	ModernUI_Checkbox Messages	57
3.3.3	ModernUI_Checkbox Properties	57
3.4	ModernUI_ProgressBar	57
3.4.1	ModernUI_ProgressBar Functions	58
3.4.2	ModernUI_ProgressBar Messages	61
3.4.3	ModernUI_ProgressBar Properties	62

3.5	ModernUI_ProgressDots	63
3.5.1	ModernUI_ProgressDots Functions	64
3.5.2	ModernUI_ProgressDots Messages	66
3.5.3	ModernUI_ProgressDots Properties	66
3.6	ModernUI_Region	66
3.6.1	ModernUI_Region Functions	66
3.6.2	ModernUI_Region Messages	71
3.6.3	ModernUI_Region Properties	71
3.7	ModernUI_SmartPanel	71
3.7.1	ModernUI_SmartPanel Functions	72
3.7.2	ModernUI_SmartPanel Messages	79
3.7.3	ModernUI_SmartPanel Properties	84
3.8	ModernUI_Spinner	85
3.8.1	ModernUI_Spinner Functions	85
3.8.2	ModernUI_Spinner Messages	93
3.8.3	ModernUI_Spinner Properties	97
3.9	ModernUI_Text	97
3.9.1	ModernUI_Text Functions	97
3.9.2	ModernUI_Text Properties	101
3.10	ModernUI_Tooltip	102
3.10.1	ModernUI_Tooltip Functions	102
3.10.2	ModernUI_Tooltip Messages	104
3.10.3	ModernUI_Tooltip Properties	104
3.11	ModernUI_TrayMenu	104
3.11.1	ModernUI_TrayMenu Functions	104
3.11.2	ModernUI_TrayMenu Messages	111
3.11.3	ModernUI_TrayMenu Properties	111
4	Advanced	113
4.1	ModernUI x86 Build	113
4.1.1	Building ModernUI Library	113
4.1.2	Building ModernUI Controls	114
4.1.3	Debug x86 Builds	114
4.2	ModernUI x64 Build	114
4.2.1	Building ModernUI Library	114
4.2.2	Building ModernUI Controls	115
4.2.3	Debug x64 Builds	115
4.3	RadASM Support	115
4.3.1	RadASM auto-complete / intellisense	116
4.3.2	RadASM Design Time ModernUI Controls	116
4.3.3	RadASM with UASM	117
4.3.4	RadASM Colorized ModernUI Library Types	118
4.4	Developers	118
5	Indices and tables	121

Contents:

ModernUI is a framework library and a collection of custom controls for win32 assembler, created to help modernize the standard win32 controls, and to add or emulate new control types and features of modern UX/UI designs and other graphical frameworks.

For years, new control types and UI designs for operating system and web have progressed, whilst older win32 base controls which are used by win32 assemblers (masm32 etc) have remained untouched. The ModernUI project has been created to help modernize these existing controls and add to them. The main goals of the project are:

- Provide a modern update to existing win32 base class visual controls.
- Add new control types to reflect newer UI/UX design influences.
- Emulate control types from UWP, WPF, .Net & Web 2.0 platforms.

Contents:

1.1 Installation ModernUI

Download the latest version of the main ModernUI library and extract the files. The latest release can be found in the Release folder, or via the releases section of the Github repository.

- ModernUI x86: <https://github.com/mrfearless/ModernUI>
- ModernUI x64: <https://github.com/mrfearless/ModernUI64>

For ModernUI x86:

- Copy the ModernUI.inc file to your MASM32\include folder (or wherever your includes are located)
- Copy the ModernUI.lib file to your MASM32\lib folder (or wherever your libraries are located)

For ModernUI x64:

- Copy the ModernUI.inc file to your UASM32\include folder (or wherever your includes are located)
- Copy the ModernUI.lib file to your UASM32\lib\x64 folder (or wherever your libraries are located)

1.2 Setup ModernUI

Add the following to your project:

```
Include ModernUI.inc  
Includelib ModernUI.lib
```

Contents:

2.1 ModernUI Base Functions

Contents:

2.1.1 MUIGetIntProperty

MUIGetIntProperty, hWin:*MUIWND*, Property:*MUIPROPERTY*

Get a specified property from the **internal** properties structure. Properties are defined as constants (eg @MyPropertyX EQU 4) and are offsets into the memory location used to store the properties value. See *MUIAllocMemProperties* for details on allocating memory for properties.

Parameters

- [in] **hWin** - handle to the ModernUI control
- [in] **Property** - The property to set

Return

Returns the property value or NULL

Example

```
.data
; Define internal properties
@MyPropertyXcoord EQU 0
@MyPropertyYcoord EQU 4
@MyPropertyString EQU 8
```

```
LOCAL dwX:DWORD
LOCAL dwY:DWORD
LOCAL lpszMyString:DWORD

Invoke MUIGetIntProperty, hMyControl, @MyPropertyXcoord
mov dwX, eax

Invoke MUIGetIntProperty, hMyControl, @MyPropertyYcoord
mov dwY, eax

Invoke MUIGetIntProperty, hMyControl, @MyPropertyString
mov lpszMyString, eax
```

See Also

[MUIAllocMemProperties](#), [MUISetIntProperty](#), [MUIGetExtProperty](#), [MUISetExtProperty](#)

2.1.2 MUISetIntProperty

MUISetIntProperty, hWin:[MUIWND](#), Property:[MUIPROPERTY](#), PropertyValue:[MUIPROPERTYVALUE](#)

Sets a specified property in the **internal** properties structure. Properties are defined as constants (eg @MyPropertyX EQU 4) and are offsets into the memory location used to store the properties value. See [MUIAllocMemProperties](#) for details on allocating memory for properties.

Parameters

- [in] **hWin** - handle to the ModernUI control
- [in] **Property** - The property to set
- [in] **PropertyValue** - The value to set the property to

Return

Returns the previously set property value or NULL

Example

```
.data
MyString db 'test',0

; Define internal properties
@MyPropertyXcoord EQU 0
@MyPropertyYcoord EQU 4
@MyPropertyString EQU 8

LOCAL dwX:DWORD
LOCAL dwY:DWORD

mov dwX, 1234
Invoke MUISetIntProperty, hMyControl, @MyPropertyXcoord, dwX

mov eax, 5678
Invoke MUISetIntProperty, hMyControl, @MyPropertyYcoord, eax

lea eax, MyString
Invoke MUISetIntProperty, hMyControl, @MyPropertyString, eax
```

See Also

MUIAllocMemProperties, MUIGetIntProperty, MUIGetExtProperty, MUISetExtProperty

2.1.3 MUIGetExtProperty

MUIGetExtProperty, hWin:*MUIWND*, Property:*MUIPROPERTY*

Get a specified property from the **external** properties structure. Properties are defined as constants (eg @MyPropertyX EQU 4) and are offsets into the memory location used to store the properties value. See *MUIAllocMemProperties* for details on allocating memory for properties.

Parameters

- [in] **hWin** - handle to the ModernUI control
- [in] **Property** - The property to set

Return

Returns the property value or NULL

Example

```
.data
; Define external properties
@MyPropertyXcoord EQU 0
@MyPropertyYcoord EQU 4
@MyPropertyString EQU 8
```

```
LOCAL dwX:DWORD
LOCAL dwY:DWORD
LOCAL lpszMyString:DWORD

Invoke MUIGetExtProperty, hMyControl, @MyPropertyXcoord
mov dwX, eax

Invoke MUIGetExtProperty, hMyControl, @MyPropertyYcoord
mov dwY, eax

Invoke MUIGetExtProperty, hMyControl, @MyPropertyString
mov lpszMyString, eax
```

See Also

MUIAllocMemProperties, MUISetExtProperty, MUIGetIntProperty, MUISetIntProperty

2.1.4 MUISetExtProperty

MUISetExtProperty, hWin:*MUIWND*, Property:*MUIPROPERTY*, PropertyValue:*MUIPROPERTYVALUE*

Sets a specified property in the **external** properties structure. Properties are defined as constants (eg @MyPropertyX EQU 4) and are offsets into the memory location used to store the properties value. See *MUIAllocMemProperties* for details on allocating memory for properties.

Parameters

- [in] **hWin** - handle to the ModernUI control
- [in] **Property** - The property to set

- [in] **PropertyValue** - The value to set the property to

Return

Returns the previously set property value or NULL

Example

```
.data
MyString db 'test',0

; Define external properties
@MyPropertyXcoord EQU 0
@MyPropertyYcoord EQU 4
@MyPropertyString EQU 8
```

```
LOCAL dwX:DWORD
LOCAL dwY:DWORD

mov dwX, 1234
Invoke MUISetExtProperty, hMyControl, @MyPropertyXcoord, dwX

mov eax, 5678
Invoke MUISetExtProperty, hMyControl, @MyPropertyYcoord, eax

lea eax, MyString
Invoke MUISetExtProperty, hMyControl, @MyPropertyString, eax
```

See Also

MUIAllocMemProperties, MUIGetExtProperty, MUIGetIntProperty, MUISetIntProperty

2.2 ModernUI DPI Functions

Contents:

2.3 ModernUI Font Functions

Contents:

2.3.1 MUIPointSizeToLogicalUnit

MUIPointSizeToLogicalUnit, hWin:*MUIWND*, PointSize:*MUIVALUE*

Convert font point size eg 12 to logical unit size for use with [CreateFont](#) or [CreateFontIndirect](#)

Parameters

- [in] **hWin** - handle to window to use dc for converting the point size to logical size
- [in] **PointSize** - a value representing the point size to convert

Return

Returns the font height for the point size specified

Example

```
LOCAL dwFontHeight:DWORD

MUIPointSizeToLogicalUnit, hWin, 12
mov dwFontHeight, eax

Invoke CreateFont, dwFontHeight, ...
```

See Also

[CreateFont](#), [CreateFontIndirect](#)

2.4 ModernUI GDI Functions

Contents:

2.4.1 MUIGDIBlend

Parameters

- 1
- 2
- 3

Return**Example**

```
**See Also**
```

<>, <>

2.4.2 MUIGDIBlendBitmaps

Parameters

- 1
- 2
- 3

Return**Example**

```
**See Also**
```

<>, <>

2.4.3 MUIGDIDoubleBufferFinish

MUIGDIDoubleBufferFinish, hdcBuffer:HDC, hBufferBitmap:HBITMAP, hBitmapUsed:HBITMAP, hFontUsed:HFONT, hBrushUsed:HBRUSH, hPenUsed:HPEN

Finishes double buffering and cleans up afterwards. Used in a WM_PAINT event. Place before EndPaint call and after all Blt (eg BitBlt) calls. **hdcBuffer** is the HDC of the double buffer (eg. hdcMem). **hBufferBitmap** is the handle to the double buffer bitmap. **hBitmapUsed**, **hFontUsed**, **hBrushUsed**, and **hPenUsed** are optional parameters. If you have used a bitmap image HBITMAP (not the double buffer bitmap which is **hBufferBitmap**) or a font HFONT, brush HBRUSH or a pen HPEN in your code in the **hdcBuffer** you can pass the handles here for cleaning up, otherwise pass NULL or 0 for those other parameters.

Parameters

- [in] **hdcBuffer** - the double buffer dc (eg hdcMem)
- [in] **hBufferBitmap** - the double buffer bitmap HBITMAP
- [in] **hBitmapUsed** - optional HBITMAP used in double buffer dc
- [in] **hFontUsed** - optional handle to a HFONT used in double buffer dc
- [in] **hBrushUsed** - optional handle to a HBRUSH used in double buffer dc
- [in] **hPenUsed** - optional handle to a HPEN used in double buffer dc

Return

None

Example

```
Invoke BitBlt, hdc, 0, 0, rect.right, rect.bottom, hdcMem, 0, 0, SRCCOPY

Invoke MUIGDIDoubleBufferFinish, hdcMem, hBufferBitmap, 0, 0, hBrush, 0

Invoke EndPaint, hWin, Addr ps
```

See Also

MUIGDIDoubleBufferStart, *BeginPaint*, *WM_PAINT*, *EndPaint*

2.4.4 MUIGDIDoubleBufferStart

MUIGDIDoubleBufferStart, hWin:MUIWND, hdcSource:HDC, lpHDCBuffer:LPHDC, lpClientRect:LPRECT, lpBufferBitmap:LPHBITMAP

Starts double buffering. Used in a WM_PAINT event. Place after BeginPaint call. **lpHDCBuffer** points to a variable used to store the HDC of the double buffer (eg. hdcMem). **lpClientRect** points to a RECT structure used to store the client rectangle. **lpBufferBitmap** points to a variable used to store the double buffer HBITMAP.

Parameters

- [in] **hWin** - handle to the window to paint. Typically the control itself
- [in] **hdcSource** - the HDC source, typically the dc returned from the BeginPaint call
- [out] **lpHDCBuffer** - pointer to the variable used to store the double buffer HDC
- [out] **lpClientRect** - pointer to the variable used to store the rectangle RECT of the window
- [out] **lpBufferBitmap** - pointer to the variable used to store the double buffer bitmap HBITMAP

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
hdc:HDC
LOCAL hdcMem:HDC
LOCAL hBufferBitmap:HBITMAP
LOCAL rect:RECT

Invoke BeginPaint, hWin, Addr ps
mov hdc, eax

Invoke MUIGDIDoubleBufferStart, hWin, hdc, Addr hdcMem, Addr rect, Addr hBufferBitmap
```

See Also

MUIGDIDoubleBufferFinish, *BeginPaint*, *WM_PAINT*, *EndPaint*

2.4.5 MUIGDIPaintFill

MUIGDIPaintFill, hdc:[HDC](#), lpFillRect:[LPRECT](#), FillColor:[MUICOLORRGB](#)

Fills a rectangle, as specified by the **lpFillRect** parameter which points to a RECT, with **FillColor** which is a specific COLORREF color, to a HDC as specified by **hdc**.

Parameters

- [in] **hdc** - dc to paint the filled rectangle to
- [in] **lpFillRect** - points to a RECT that defines area to paint fill
- [in] **FillColor** - color to paint fill

Return

None

Example

```
LOCAL rect:RECT
LOCAL BackColor:COLORREF

Invoke GetClientRect, hWin, Addr rect
mov BackColor, MUI_RGBCOLOR(127, 200, 240)

Invoke MUIGDIPaintFill, hdc, Addr rect, BackColor
```

See Also

MUIGDIPaintFrame, *MUI_RGBCOLOR*

2.4.6 MUIGDIPaintFrame

MUIGDIPaintFrame, hdc:[HDC](#), lpFrameRect:[LPRECT](#), FrameColor:[MUICOLORRGB](#), FrameStyle:[MUIPFS](#)

Draws a border (or parts of) around a rectangle, as specified by the **lpFrameRect** parameter which points to a RECT, with **FrameColor** which is a specific COLORREF color, to a HDC as specified by **hdc**.

Parameters

- [in] **hdc** - dc to paint the frame to
- [in] **lpFrameRect** - points to a RECT that defines area to paint the frame
- [in] **FrameColor** - color to paint the frame edges
- [in] **FrameStyle** - indicates what parts of the frame are painted. **FrameStyle** can be a combination of the following flags: MUIPFS_NONE, MUIPFS_LEFT, MUIPFS_TOP, MUIPFS_BOTTOM, MUIPFS_RIGHT or MUIPFS_ALL

Return

None

Example

```
LOCAL rect:RECT
LOCAL BorderColor:COLORREF

Invoke GetClientRect, hWin, Addr rect
mov BorderColor, MUI_RGBCOLOR(127, 200, 240)

Invoke MUIGDIPaintFrame, hdc, Addr rect, BorderColor, MUIPFS_ALL
```

See Also

MUIGDIPaintFill, MUI_RGBCOLOR

2.4.7 MUIGDIRotateCenterBitmap

Parameters

- 1
- 2
- 3

Return**Example**

```
**See Also**
```

<>, <>

2.4.8 MUIGDISTretchBitmap

Parameters

- 1
- 2
- 3

Return**Example**

```
**See Also**
```

<>, <>

2.4.9 MUIGDISTretchImage

Parameters

- 1
- 2
- 3

Return

Example

```
**See Also**
```

<>, <>

2.5 ModernUI GDIPlus Functions

Contents:

2.5.1 MUIGDIPlusDoubleBufferFinish

MUIGDIPlusDoubleBufferFinish, hWin:*MUIWND*, pGraphics:*GPGGRAPHICS*, pBitmap:*GPIMAGE*, pGraphics-Buffer:*GPGGRAPHICS*

Finish Double Buffering for **GDI+** & copy finished pGraphicsBuffer to pGraphics (HDC). Used in a **WM_PAINT** event. Place before **EndPaint**

Parameters

- [in] **hWin** - handle to window that has double buffering (gdi+)
- [in] **pGraphics** - graphics context for the **hWin** window
- [in] **hBitmap** - double buffer image
- [in] **pGraphicsBuffer** - double buffer graphics context

Return

None

Example

```
LOCAL hdc:HDC
LOCAL pGraphics:DWORD
LOCAL pGraphicsBuffer:DWORD
LOCAL pBitmap:DWORD

Invoke BeginPaint, hWin, Addr ps
mov hdc, eax

mov pGraphics, 0
mov pGraphicsBuffer, 0
```

(continues on next page)

(continued from previous page)

```

mov pBitmap, 0

Invoke GdiCreateFromHDC, hdc, Addr pGraphics

; Start GDI+ Double Buffer
Invoke MUIGDIPlusDoubleBufferStart, hWin, pGraphics, Addr pBitmap, Addr_
↳pGraphicsBuffer

; Draw with GDI+ to pGraphicsBuffer context

; Finish GDI+ Double Buffer
Invoke MUIGDIPlusDoubleBufferFinish, hWin, pGraphics, pBitmap, pGraphicsBuffer

.IF pGraphics != 0
    Invoke GdiDeleteGraphics, pGraphics
.ENDIF

Invoke EndPaint, hWin, Addr ps

```

See Also*MUIGDIPlusDoubleBufferStart, WM_PAINT, EndPaint***2.5.2 MUIGDIPlusDoubleBufferStart**

MUIGDIPlusDoubleBufferStart, hWin:*MUIWND*, pGraphics:*GPGRAPHICS*, lpBitmapHandle:*LPGPIMAGE*, lpGraphicsBuffer:*LPGPGGRAPHICS*

Start Double Buffering for **GDI+**. Used in a **WM_PAINT** event. Place after *BeginPaint*

Parameters

- [in] **hWin** - handle to window to start double buffering (gdi+) for
- [in] **pGraphics** - graphics context for the **hWin** window
- [out] **lpBitmapHandle** - pointer to variable to store double buffer image
- [out] **lpGraphicsBuffer** - pointer to variable to store double buffer graphics context

Return

None

Example

```

LOCAL hdc:HDC
LOCAL pGraphics:DWORD
LOCAL pGraphicsBuffer:DWORD
LOCAL pBitmap:DWORD

Invoke BeginPaint, hWin, Addr ps
mov hdc, eax

mov pGraphics, 0
mov pGraphicsBuffer, 0
mov pBitmap, 0

Invoke GdiCreateFromHDC, hdc, Addr pGraphics

```

(continues on next page)

(continued from previous page)

```

; Start GDI+ Double Buffer
Invoke MUIGDIPlusDoubleBufferStart, hWin, pGraphics, Addr pBitmap, Addr_
↳pGraphicsBuffer

; Draw with GDI+ to pGraphicsBuffer context

; Finish GDI+ Double Buffer
Invoke MUIGDIPlusDoubleBufferFinish, hWin, pGraphics, pBitmap, pGraphicsBuffer

.IF pGraphics != 0
    Invoke GdipDeleteGraphics, pGraphics
.ENDIF

Invoke EndPaint, hWin, Addr ps

```

See Also

MUIGDIPlusDoubleBufferFinish, *WM_PAINT*, *BeginPaint*

2.5.3 MUIGDIPlusFinish

MUIGDIPlusFinish

Finish ModernUI **GDI+** framework (wrapper for *GdiplusShutdown*). Placed after *WinMain* call before *ExitProcess* or during exit of a ModernUI control during a *WM_DESTROY* or a *WM_NCDESTROY* event

Parameters

None

Return

None

Example

```
Invoke MUIGDIPlusFinish
```

```

Invoke MUIGDIPlusStart ; Start GDI+ before program starts
Invoke WinMain, hInstance, NULL, CommandLine, SW_SHOWDEFAULT
Invoke MUIGDIPlusFinish ; Finish GDI+ as program exits
Invoke ExitProcess, eax

```

```

.ELSEIF eax == WM_CREATE
    Invoke MUIAllocMemProperties, hWin, MUI_INTERNAL_PROPERTIES, SIZEOF _MUI_MYCONTROL_
    ↳PROPERTIES
    Invoke MUIAllocMemProperties, hWin, MUI_EXTERNAL_PROPERTIES, SIZEOF MUI_MYCONTROL_
    ↳PROPERTIES
    Invoke MUIGDIPlusStart ; Start GDI+
    Invoke _MUI_MyControlInit, hWin
    mov eax, 0
    ret

.ELSEIF eax == WM_NCDESTROY
    Invoke _MUI_MyControlCleanup, hWin
    Invoke MUIFreeMemProperties, hWin, MUI_INTERNAL_PROPERTIES

```

(continues on next page)

(continued from previous page)

```

Invoke MUIFreeMemProperties, hWin, MUI_EXTERNAL_PROPERTIES
Invoke MUIGDIPlusFinish ; Finish GDI+
mov eax, 0
ret

```

See Also

MUIGDIPlusStart, *WM_DESTROY*, *WM_NCDESTROY*, *GdiplusShutdown*

2.5.4 MUIGDIPlusPaintFill

MUIGDIPlusPaintFill, pGraphics:*GPGRAPHICS*, lpFillGdipRect:*LPGPRECT*, FillColor:*MUICOLORARGB*

Fills a rectangle, as specified by the **lpFillRect** parameter which points to a GDIPRECT, with **FillColor** which is a specific ARGB color, to a graphics context as specified by **pGraphics**. Note: a RECT can be converted to the *GDIPRECT* using the *MUIGDIPlusRectToGdipRect* function.

Parameters

- [in] **pGraphics** - graphics context to paint the filled rectangle to
- [in] **lpFillRect** - points to a GDIPRECT that defines area to paint fill
- [in] **FillColor** - ARGB color to paint fill

Return

None

Example

```

GDIPRECT    STRUCT
    left     REAL4 ?
    top      REAL4 ?
    right    REAL4 ?
    bottom   REAL4 ?
GDIPRECT    ENDS

```

```

LOCAL hdc:HDC
LOCAL pGraphics:DWORD
LOCAL rect:RECT
LOCAL gdiprect:GDIPRECT
LOCAL BackColor:DWORD

```

```

Invoke BeginPaint, hWin, Addr ps
mov hdc, eax

```

```

mov pGraphics, 0
Invoke GdipCreateFromHDC, hdc, Addr pGraphics

```

```

Invoke GetClientRect, hWin, Addr rect
Invoke MUIGDIPlusRectToGdipRect, Addr rect, Addr gdiprect

```

```

mov BackColor, MUI_ARGBCOLOR(255, 127, 200, 240)

```

```

Invoke MUIGDIPlusPaintFill, pGraphics, Addr gdiprect, BackColor

```

See Also

MUIGDIPlusPaintFillI, MUIGDIPlusPaintFrame, MUIGDIPlusPaintFrameI, MUI_ARGBCOLOR, MUIGDIPlusRectToGdipRect, GDIPRECT

2.5.5 MUIGDIPlusPaintFillI

MUIGDIPlusPaintFillI, pGraphics:*GPGRAPHICS*, lpFillRectI:*LPRECT*, FillColor:*MUICOLORARGB*

Fills a rectangle, as specified by the **lpFillRectI** parameter which points to a RECT, with **FillColor** which is a specific ARGB color, to a graphics context as specified by **pGraphics**.

Parameters

- [in] **pGraphics** - graphics context to paint the filled rectangle to
- [in] **lpFillRectI** - points to a RECT that defines area to paint fill
- [in] **FillColor** - ARGB color to paint fill

Return

None

Example

```
LOCAL hdc:HDC
LOCAL pGraphics:DWORD
LOCAL rect:RECT
LOCAL BackColor:DWORD

Invoke BeginPaint, hWin, Addr ps
mov hdc, eax

mov pGraphics, 0
Invoke GdipCreateFromHDC, hdc, Addr pGraphics

Invoke GetClientRect, hWin, Addr rect
mov BackColor, MUI_ARGBCOLOR(255, 127, 200, 240)

Invoke MUIGDIPlusPaintFillI, pGraphics, Addr rect, BackColor
```

See Also

MUIGDIPlusPaintFill, MUIGDIPlusPaintFrameI, MUIGDIPlusPaintFrame, MUI_ARGBCOLOR

2.5.6 MUIGDIPlusPaintFrame

MUIGDIPlusPaintFrame, pGraphics:*GPGRAPHICS*, lpFrameGdipRect:*LPGPRECT*, FrameColor:*MUICOLORARGB*, FrameStyle:*MUIPFS*

Draws a border (or parts of) around a rectangle, as specified by the **lpFrameRect** parameter which points to a GDIPRECT, with **FrameColor** which is a specific ARGB color, to a graphics context as specified by **pGraphics**. Note: a RECT can be converted to the *GDIPRECT* using the *MUIGDIPlusRectToGdipRect* function.

Parameters

- [in] **pGraphics** - graphics context to paint the frame to
- [in] **lpFrameRect** - points to a GDIPRECT that defines area to paint the frame
- [in] **FrameColor** - ARGB color to paint the frame edges

- [in] **FrameStyle** - indicates what parts of the frame are painted. **FrameStyle** can be a combination of the following flags: MUIPFS_NONE, MUIPFS_LEFT, MUIPFS_TOP, MUIPFS_BOTTOM, MUIPFS_RIGHT or MUIPFS_ALL

Return

None

Example

```
GDIPRECT    STRUCT
    left     REAL4 ?
    top      REAL4 ?
    right    REAL4 ?
    bottom   REAL4 ?
GDIPRECT    ENDS
```

```
LOCAL hdc:HDC
LOCAL pGraphics:DWORD
LOCAL rect:RECT
LOCAL gdiprect:GDIPRECT
LOCAL BorderColor:DWORD

Invoke BeginPaint, hWin, Addr ps
mov hdc, eax

mov pGraphics, 0
Invoke GdiCreateFromHDC, hdc, Addr pGraphics

Invoke GetClientRect, hWin, Addr rect
Invoke MUIGDIPlusRectToGdiRect, Addr rect, Addr gdiprect

mov BorderColor, MUI_ARGBCOLOR(255, 48, 48, 48)

Invoke MUIGDIPlusPaintFrame, pGraphics, Addr gdiprect, BorderColor, MUIPFS_ALL
```

See Also

MUIGDIPlusPaintFrameI, MUIGDIPlusPaintFill, MUIGDIPlusPaintFillI, MUI_ARGBCOLOR, MUIGDIPlusRectToGdiRect, GDIPRECT

2.5.7 MUIGDIPlusPaintFrameI

MUIGDIPlusPaintFrameI, pGraphics:*GPGRAPHS*, lpFrameRectI:*LPRECT*, FrameColor:*MUICOLORARGB*, FrameStyle:*MUIPFS*

Draws a border (or parts of) around a rectangle, as specified by the **lpFrameRectI** parameter which points to a RECT, with **FrameColor** which is a specific ARGB color, to a graphics context as specified by **pGraphics**.

Parameters

- [in] **pGraphics** - graphics context to paint the frame to
- [in] **lpFrameRectI** - points to a RECT that defines area to paint the frame
- [in] **FrameColor** - ARGB color to paint the frame edges
- [in] **FrameStyle** - indicates what parts of the frame are painted. **FrameStyle** can be a combination of the following flags: MUIPFS_NONE, MUIPFS_LEFT, MUIPFS_TOP, MUIPFS_BOTTOM, MUIPFS_RIGHT or MUIPFS_ALL

Return

None

Example

```

LOCAL hdc:HDC
LOCAL pGraphics:DWORD
LOCAL rect:RECT
LOCAL BorderColor:DWORD

Invoke BeginPaint, hWin, Addr ps
mov hdc, eax

mov pGraphics, 0
Invoke GdipCreateFromHDC, hdc, Addr pGraphics

Invoke GetClientRect, hWin, Addr rect
mov BorderColor, MUI_ARGBCOLOR(255, 48, 48, 48)

Invoke MUIGDIPlusPaintFrameI, pGraphics, Addr rect, BorderColor, MUIPFS_ALL

```

See Also

MUIGDIPlusPaintFrame, MUIGDIPlusPaintFillI, MUIGDIPlusPaintFill, MUI_ARGBCOLOR

2.5.8 MUIGDIPlusRectToGdipRect

MUIGDIPlusRectToGdipRect, lpRect:*LPRECT*, lpGdipRect:*LPGPRECT*

Convert a RECT structure to a GDIPRECT structure.

Parameters

- [in] **lpRect** - pointer to RECT to convert
- [out] **lpGdipRect** - pointer to GDIRECT to store the result

Return

None

Example

```

GDIPRECT    STRUCT
    left     REAL4 ?
    top      REAL4 ?
    right    REAL4 ?
    bottom   REAL4 ?
GDIPRECT    ENDS

```

```

LOCAL rect:RECT
LOCAL gdirect:GDIRECT

Invoke GetClientRect, hWin, Addr rect
Invoke MUIGDIPlusRectToGdipRect, Addr rect, Addr gdirect

```

See Also

MUIGDIPlusPaintFill, MUIGDIPlusPaintFrame, GDIPRECT

2.5.9 MUIGDIPlusRotateCenterImage

MUIGDIPlusRotateCenterImage, hImage:*GPIMAGE*, fAngle:REAL4

Rotates a **GDI+** image **hImage** around its center. **fAngle** is a REAL4 value indicating the angle in degrees to rotate the original **hImage** by.

Parameters

- [in] **hImage** - handle to a GDI+ image to rotate
- [in] **fAngle** - degrees to rotate image

Return

Returns a new image handle of the rotated image

Example

```
IFDEF FP4
    FP4 MACRO value
    LOCAL vname
    .data
    align 4
        vname REAL4 value
    .code
    EXITM <vname>
    ENDM
ENDIF
```

```
Invoke MUIGDIPlusRotateCenterImage, pBitmap, FP4(45.0)
```

See Also

MUIGDIPlusDoubleBufferStart, MUIGDIPlusDoubleBufferFinish

2.5.10 MUIGDIPlusStart

MUIGDIPlusStart

Start of ModernUI **GDI+** framework (wrapper for *GdiplusStartup*). Placed at start of program before WinMain call or during creation of a ModernUI control during a **WM_CREATE** event.

Parameters

None

Return

None

Example

```
Invoke MUIGDIPlusStart
```

```
Invoke MUIGDIPlusStart ; Start GDI+ before program starts
Invoke WinMain, hInstance, NULL, CommandLine, SW_SHOWDEFAULT
Invoke MUIGDIPlusFinish ; Finish GDI+ as program exits
Invoke ExitProcess, eax
```

```
.ELSEIF eax == WM_CREATE
    Invoke MUIAllocMemProperties, hWin, MUI_INTERNAL_PROPERTIES, SIZEOF _MUI_MYCONTROL_
    ↳PROPERTIES
    Invoke MUIAllocMemProperties, hWin, MUI_EXTERNAL_PROPERTIES, SIZEOF MUI_MYCONTROL_
    ↳PROPERTIES
    Invoke MUIGDIPlusStart ; Start GDI+
    Invoke _MUI_MyControlInit, hWin
    mov eax, 0
    ret

.ELSEIF eax == WM_NCDESTROY
    Invoke _MUI_MyControlCleanup, hWin
    Invoke MUIFreeMemProperties, hWin, MUI_INTERNAL_PROPERTIES
    Invoke MUIFreeMemProperties, hWin, MUI_EXTERNAL_PROPERTIES
    Invoke MUIGDIPlusFinish ; Finish GDI+
    mov eax, 0
    ret
```

See Also

MUIGDIPlusFinish, *WM_CREATE*, *GdiplusStartup*

2.5.11 MUILoadPngFromResource**Parameters**

- 1
- 2
- 3

Return**Example**

```
**See Also**
```

◇, ◇

2.6 ModernUI Image Functions

Contents:

2.6.1 MUICreateBitmapFromMemory

MUICreateBitmapFromMemory, pBitmapData: *POINTER*

Create an bitmap from data stored in memory. Typically defined in the `.data` section as a variable with a sequence of binary data. Use `bin2dbx` from `masm32` package to generate the byte sequence required. **pBitmapData** is a pointer to the bitmap file data.

Parameters

- [in] **pBitmapData** - pointer to the bitmap file data

Return

Returns a bitmap handle (HBITMAP) on success

Example

```
.data
BmpData db 66,77,58,0,0,0,0,0,0,0,54,0,0,0,40,0
          db 0,0,1,0,0,0,1,0,0,0,1,0,24,0,0,0
          db 0,0,0,0,0,0,194,30,0,0,194,30,0,0,0,0
          db 0,0,0,0,0,0,128,128,128,0
```

```
LOCAL hBmp

Invoke MUICreateBitmapFromMemory, Addr BmpData
.IF eax == NULL
    ; error
.ENDIF
mov hBmp, eax
```

See Also

MUICreateIconFromMemory, MUICreateCursorFromMemory

2.6.2 MUICreateCursorFromMemory

MUICreateCursorFromMemory, pCursorData:*POINTER*

Creates a cursor from icon/cursor data stored in memory. Typically defined in the `.data` section as a variable with a sequence of binary data. Use `bin2dbex` from `masm32` package to generate the byte sequence required. **pCursorData** is a pointer to the cursor file data.

Parameters

- [in] **pCursorData**- pointer to the cursor file data

Return

Returns a cursor (a special animated icon) handle (HICON) on success

Example

```
.data
CursorData db 0,0,1,0,1,0,1,1,0,0,1,0,24,0,48,0
            db 0,0,22,0,0,0,40,0,0,0,1,0,0,0,2,0
            db 0,0,1,0,24,0,0,0,0,0,8,0,0,0,0,0
            db 0,0,0,0,0,0,0,0,0,0,0,0,0,128,128
            db 128,11,0,0,0,0
```

```
LOCAL hCursor

Invoke MUICreateCursorFromMemory, Addr CursorData
.IF eax == NULL
    ; error
.ENDIF
mov hCursor, eax
```

See Also

MUICreateBitmapFromMemory, MUICreateIconFromMemory

2.6.3 MUICreateIconFromMemory

MUICreateIconFromMemory, pIconData:*POINTER*, iIcon:*MUIVALUE*

Create an icon from data stored in memory. Typically defined in the `.data` section as a variable with a sequence of binary data. Use `bin2dbex` from `masm32` package to generate the byte sequence required. **pIconData** is a pointer to the icon file data.

Parameters

- [in] **pIconData** - pointer to the icon file data
- [in] **iIcon** - set to 0

Return

Returns an icon handle (HICON) on success

Example

```
.data
IconData db 0,0,1,0,1,0,1,1,0,0,1,0,24,0,48,0
          db 0,0,22,0,0,0,40,0,0,0,1,0,0,0,2,0
          db 0,0,1,0,24,0,0,0,0,0,8,0,0,0,0,0
          db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,128
          db 128,11,0,0,0,0
```

```
LOCAL hIco

Invoke MUICreateIconFromMemory, Addr IconData, 0
.IF eax == NULL
    ; error
.ENDIF
mov hIco, eax
```

See Also

MUICreateBitmapFromMemory, MUICreateCursorFromMemory

2.6.4 MUIGetImageSize

MUIGetImageSize, hImage:*MUIIMAGE*, ImageHandleType:*MUIIT*, lpImageWidth:*LPMUIVALUE*, lpImageHeight:*LPMUIVALUE*

Gets the image size of the specified image, based on the specified type: icon, bitmap or png. Width and Height are returned into the variables pointed to by the `lpdwImageWidth` and `lpdwImageHeight` parameters if successful.

Parameters

- [in] **hImage** - handle to image to get dimensions for
- [in] **ImageHandleType** - type of image used in **hImage**: `MUIIT_NONE`, `MUIIT_BMP`, `MUIIT_ICO`, or `MUIIT_PNG`
- [out] **lpImageWidth** - width of image
- [out] **lpImageHeight** - height of image

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUIGetImageSize, hMyBitmap, MUIIT_BMP, Addr dwWidth, Addr dwHeight
```

See Also

MUIGetImageSizeEx

2.6.5 MUIGetImageSizeEx

MUIGetImageSizeEx, hWin:*MUIWND*, hImage:*MUIIMAGE*, ImageHandleType:*MUIIT*, lpImageWidth:*LPMUIVALUE*, lpImageHeight:*LPMUIVALUE*, lpImageX:*LPMUIVALUE*, lpImageY:*LPMUIVALUE*

Similar to MUIGetImageSize, but also returns centering x and y co-ordinate information based on rectangle of hWin

Parameters

- [in] **hWin** - handle to window to get center x and y coords for
- [in] **hImage** - handle to image to get dimensions for
- [in] **ImageHandleType** - type of image used in **hImage**: MUIIT_NONE, MUIIT_BMP, MUIIT_ICO, or MUIIT_PNG
- [out] **lpImageWidth** - width of image
- [out] **lpImageHeight** - height of image
- [out] **lpImageX** - x coord of image centered in hWin
- [out] **lpImageY** - y coord of image centered in hWin

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUIGetImageSizeEx, hWin, hMyBitmap, MUIIT_BMP, Addr dwWidth, Addr dwHeight, └  
└Addr xpos, Addr ypos
```

See Also

MUIGetImageSize

2.6.6 MUILoadBitmapFromResource

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Example

```
**See Also**
```

◁, ▷

2.6.7 MUILoadIconFromResource

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Example

```
**See Also**
```

<>, <>

2.6.8 MUILoadImageFromResource

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Example

```
**See Also**
```

<>, <>

2.7 ModernUI Memory Functions

Contents:

2.7.1 MUIAllocMemProperties

MUIAllocMemProperties, hWin:[MUIWND](#), cbWndExtraOffset:[MUIPROPERTIES](#), SizeToAllocate:[MUIVALUE](#)

Allocates memory for the internal or external properties storage - typically during [WM_CREATE](#) event and this memory is freed by a call to the [MUIFreeMemProperties](#) function during [WM_DESTROY](#) or [WM_NCDESTROY](#) event.

Parameters

- [in] **hWin** - handle to the ModernUI control
- [in] **cbWndExtraOffset** - offset in the [WNDCLASS.cbWndExtra](#) memory location. For ease of use, the following constants can be used: [MUI_INTERNAL_PROPERTIES](#) or [MUI_EXTERNAL_PROPERTIES](#)
- [in] **SizeToAllocate** - size of memory block

Return

Returns TRUE if successful or FALSE otherwise

Example

```
; MyControl.inc:

; external properties for user of control
; defined in public user MyControl.inc file
@MyControlTextColor EQU 0
@MyControlBackColor EQU 4
@MyControlSpecial EQU 8
```

```
; MyControl.asm:

; external properties storage for control
; defined inside developer MyControl.asm source file
MUI_MYCONTROL STRUCT
    dwTextColor DD ?
    dwBackColor DD ?
    bSpecial DD ?
MUI_MYCONTROL ENDS

; internal properties storage for control
; defined inside developer MyControl.asm source file
_MUI_MYCONTROL STRUCT
    dwEnabled DD ?
    dwMouseOver DD ?
_MUI_MYCONTROL ENDS

; Define internal properties
; for developer use in the MyControl.asm source file
@MyControlEnabled EQU 0
@MyControlMouseOver EQU 4
```

```
; MyControl.asm:

.ELSEIF eax == WM_CREATE
    ; Alloc internal and external property storage for MyControl
    Invoke MUIAllocMemProperties, MUI_INTERNAL_PROPERTIES, SIZEOF _MUI_MYCONTROL
    Invoke MUIAllocMemProperties, MUI_EXTERNAL_PROPERTIES, SIZEOF MUI_MYCONTROL
    mov eax, TRUE
    ret
```

See Also

[MUIFreeMemProperties](#), [MUIAllocStructureMemory](#), [WM_CREATE](#), [WM_DESTROY](#), [WM_NCDESTROY](#), [WNDCLASS](#)

2.7.2 MUIFreeMemProperties

MUIFreeMemProperties, hWin:[MUIWND](#), cbWndExtraOffset:[MUIPROPERTIES](#)

Frees memory allocated via the [MUIAllocMemProperties](#) function. Used at the [WM_DESTROY](#) or [WM_NCDESTROY](#) event. Make sure any cleanup routines are called before freeing any memory.

Parameters

- [in] **hControl** - handle to the ModernUI control
- [in] **cbWndExtraOffset** - offset in the `WNDCLASS.cbWndExtra` memory location. For ease of use, the following constants can be used: `MUI_INTERNAL_PROPERTIES` or `MUI_EXTERNAL_PROPERTIES`

Return

Returns TRUE if successful or FALSE otherwise

Example

```
; MyControl.asm:

.ELSEIF eax == WM_DESTROY
    ; Cleanup before freeing memory
    Invoke MyControlCleanup, hWin
    ; Free internal and external property storage memory for MyControl
    Invoke MUIFreeMemProperties, MUI_INTERNAL_PROPERTIES
    Invoke MUIFreeMemProperties, MUI_EXTERNAL_PROPERTIES
    mov eax, 0
    ret
```

See Also

[MUIAllocMemProperties](#), [MUIAllocStructureMemory](#), `WM_DESTROY`, `WM_NCDESTROY`, `WNDCLASS`

2.7.3 MUIAllocStructureMemory

`MUIAllocStructureMemory`, `PtrStructMem`:*POINTER*, `TotalItems`:*MUIVALUE*, `ItemSize`:*MUIVALUE*

Dynamically allocates (or reallocates) memory for a specified array of structures and auto increments the total items in the array, and returns a pointer to the new current item in the array of structures.

Parameters

- [inout] **lpPtrStructMem** - pointer to variable used to store the pointer to the allocated structure
- [in] **TotalItems** - number of item elements currently in the structure
- [in] **ItemSize** - size of an item element in bytes

Notes

lpPtrStructMem is an address to receive the pointer to memory location of the base structure in memory. **lpPtrStructMem** can be NULL if **TotalItems** is 0, otherwise it must contain the address of the base structure in memory if the memory is to be increased (**TotalItems** > 0)

`ItemSize` is typically the `SIZEOF` structure to be allocated. This function calculates for you: *TotalItems * ItemSize*

If **lpPtrStructMem** is NULL then memory object is initialized to the size of *TotalItems * ItemSize* and the pointer to the memory allocated is returned

Return

Returns the pointer to the new structure item or -1 if there was a problem allocating memory

Example

```
; Widget element
Widget      STRUCT
    dwPrice  DD ?
    dwHeight DD ?
```

(continues on next page)

(continued from previous page)

```
    dwWidth    DD ?
Widget        ENDS

; pointers
pMyWidgetArray DD 0
pCurrentWidget DD 0

; Widget count
dwTotalWidgets DD 0
```

```
; Assume dwTotalWidgets was previously set to 10 beforehand

; Add a new Widget to array of Widgets
Invoke MUIAllocStructureMemory, Addr pMyWidgetArray, dwTotalWidgets, SIZEOF Widget
.IF eax == -1
    ; Error - tell user?
    ret
.ENDIF

; Save returned value as pointer to newly added element (Widget) in array
mov pCurrentWidget, eax
inc dwTotalWidgets ; update the total widgets count now

; total widgets will now be 11
; pCurrentWidget will point to this new 11th Widget
```

See Also

MUIAllocMemProperties, *MUIFreeMemProperties*

2.8 ModernUI Painting Functions

Contents:

2.8.1 MUIGetParentBackgroundBitmap

MUIGetParentBackgroundBitmap, hWin:*MUIWND*

Gets parent's background bitmap from the parent dc, at the child's location and size. For use in setting background of child to 'transparent'.

Parameters

- [in] **hWin** - handle to the window to get the parent bitmap for

Return

Returns a HBITMAP if successful or NULL otherwise

Example

```
LOCAL hBackBitmap:HBITMAP

Invoke MUIGetParentBackgroundBitmap, hWin
.IF eax != NULL
```

(continues on next page)

(continued from previous page)

```

    mov hBackBitmap, eax
    ; Store bitmap and use to paint our control's background to fake transparency
.ENDIF

```

See Also*MUIGetParentBackgroundColor***2.8.2 MUIGetParentBackgroundColor**MUIGetParentBackgroundColor, hWin:*MUIWND*

Gets a parent window background color. This function is useful for certain controls to retrieve the parent's background color and then to set their own background color based on the same value.

Parameters

- [in] **hWin** - handle to the window to get the parent color for

Return

Returns COLORREF or -1 if a null brush is set for the window

Example

```

LOCAL BackColor:COLORREF

Invoke MUIGetParentBackgroundColor, hWin
.IF eax != -1
    mov BackColor, eax
    ; Set controls own background based on parent's color
    Invoke MUISetExtProperty, hWin, @MyControlBackColor, BackColor
.ELSE
    ; Set default color if we couldn't get parents background color
    Invoke MUISetExtProperty, hWin, @MyControlBackColor, MUI_RGBCOLOR(240,240,240)
.ENDIF

```

See Also

:ref:MUIGetParentBackgroundBitmap'<MUIGetParentBackgroundBitmap>', *MUI_RGBCOLOR*

2.8.3 MUIPaintBackgroundMUIPaintBackground, hWin:*MUIWND*, Backcolor:*MUICOLORRGB*, BorderColor:*MUICOLORRGB*

Paint the background of the a window a specified COLORREF color. Optionally provide **BorderColor** for a border COLORREF color to draw. If **BorderColor** = 0, no border is drawn. If you require black for border, use 1, or MUI_RGBCOLOR(1, 1, 1)

Parameters

- [in] **hWin** - handle to window to paint background for
- [in] **Backcolor** - color to paint background with
- [in] **BorderColor** - color to paint border with

Return

None

Notes

You should handle the `WM_ERASEBKGD` and `WM_PAINT` events if you are going to use this function.

If you are using this on a window/dialog that does not use the ModernUI_CaptionBar control AND window/dialog is resizable, you should place a call to `InvalidateRect` in the `WM_NCCALCSIZE` handler to prevent ugly drawing artefacts when border is drawn whilst resize of window/dialog occurs.

The ModernUI_CaptionBar handles this call to `WM_NCCALCSIZE` already by default

Here is an example of what to include if you need:

```
.ELSEIF eax == WM_NCCALCSIZE
    Invoke InvalidateRect, hWin, NULL, TRUE
```

Example

```
.ELSEIF eax == WM_ERASEBKGD
    mov eax, 1
    ret

.ELSEIF eax == WM_PAINT
    Invoke MUIPaintBackground, hWin, MUI_RGBCOLOR(255,255,255), MUI_RGBCOLOR(48,48,48)
    mov eax, 0
    ret
```

See Also

MUIPaintBackgroundImage, *MUI_RGBCOLOR*, *WM_ERASEBKGD*, *WM_PAINT*, *WM_NCCALCSIZE*, *InvalidateRect*

2.8.4 MUIPaintBackgroundImage

MUIPaintBackgroundImage, hWin:*MUIWND*, BackColor:*MUICOLORRGB*, BorderColor:*MUICOLORRGB*, hImage:*MUIIMAGE*, ImageHandleType:*MUIIT*, ImageLocation:*MUIIL*

Same as MUIPaintBackground, but with an image.

Parameters

- [in] **hWin** - handle to window to paint background for
- [in] **BackColor** - color to paint background with
- [in] **BorderColor** - color to paint border with
- [in] **hImage** - handle to image to paint on background
- [in] **ImageHandleType** - type of image used in **hImage**: `MUIIT_NONE`, `MUIIT_BMP`, `MUIIT_ICO`, or `MUIIT_PNG`
- [in] **ImageLocation** - location where to paint **hImage**: `MUIIL_CENTER`, `MUIIL_BOTTOMLEFT`, `MUIIL_BOTTOMRIGHT`, `MUIIL_TOPLEFT`, `MUIIL_TOPRIGHT`, `MUIIL_TOPCENTER`, or `MUIIL_BOTTOMCENTER`

Return

None

Notes

You should handle the `WM_ERASEBKGD` and `WM_PAINT` events if you are going to use this function.

Example

```
.ELSEIF eax == WM_ERASEBKGD
    mov eax, 1
    ret

.ELSEIF eax == WM_PAINT
    Invoke MUIPaintBackgroundImage, hWin, MUI_RGBCOLOR(255,255,255), MUI_RGBCOLOR(48,
↪48,48), hMyBitmap, MUIIT_BMP, MUIIL_CENTER
    mov eax, 0
    ret
```

See Also

MUIPaintBackground, *MUI_RGBCOLOR*, *WM_ERASEBKGD*, *WM_PAINT*

2.9 ModernUI Region Functions

Contents:

2.9.1 MUILoadRegionFromResource

`MUILoadRegionFromResource`, `hInst:HINSTANCE`, `idRgnRes:RESID`, `lpRegionData:POINTER`, `lpSizeRegionData:LPMUIVALUE`

Loads region data from a resource, stored as `RC_DATA`.

Parameters

- [in] **hInst** - instance of the exe/dll to use for loading resources
- [in] **idRgnRes** - resource id of the region `RC_DATA` resource to load
- [out] **lpRegionData** - pointer to a variable to store the loaded region
- [out] **lpSizeRegion** - pointer to a variable to store the size of the region data

Return

Returns `TRUE` if successful, or `FALSE` otherwise

Example

```
LOCAL hinstance:DWORD
LOCAL ptrRegionData:DWORD
LOCAL dwRegionDataSize:DWORD
LOCAL hRgn:DWORD

Invoke GetModuleHandle, NULL
mov hinstance, eax

; Load region data from resource
Invoke MUILoadRegionFromResource, hinstance, idRgnRes, Addr ptrRegionData, Addr_
↪dwRegionDataSize
```

(continues on next page)

(continued from previous page)

```

; clear existing region if any from window
Invoke SetWindowRgn, hWin, NULL, FALSE

; create region based on our loaded region data
Invoke ExtCreateRegion, NULL, dwRegionDataSize, ptrRegionData
mov hRgn, eax
.IF eax == NULL
    ; error couldnt create region so exit
    ret
.ENDIF

; Set window to our newly created region (from our loaded region data)
Invoke SetWindowRgn, hWin, hRgn, TRUE

```

See Also*MUISetRegionFromResource***2.9.2 MUISetRegionFromResource**

MUISetRegionFromResource, hWin:*MUIWND*, idRgnRes:*RESID*, lpCopyRgnHandle:*LPMUIVALUE*, bRedraw:BOOL

Sets a window/controls region from a region stored as an RC_DATA resource: **idRgnRes**. If **lpdwCopyRgn** is not NULL a copy of region handle is provided (for any future calls to *FrameRgn* for example). This function internally calls the *MUILoadRegionFromResource* function.

Parameters

- [in] **hWin** - handle to the window to set a region for
- [in] **idRgnRes** - resource id of the region RC_DATA resource to load
- [out] **lpCopyRgn** - pointer to a variable to store a copy of the loaded region
- [in] **bRedraw** - redraw the window after the region is set (TRUE or FALSE)

Return

Returns TRUE if successful, or FALSE otherwise

Example

```

.const
RGN_MAP_EUROPE EQU 300 ; resource stored as id 300 in RC_DATA format

```

```

LOCAL hRegCopyMapEurope:DWORD

```

```

Invoke MUISetRegionFromResource, hWin, RGN_MAP_EUROPE, Addr hRegCopyMapEurope, TRUE
; hWin should now be shaped like a map of europe

```

See Also*MUILoadRegionFromResource*, *FrameRgn*

2.10 ModernUI Window Functions

Contents:

2.10.1 MUIApplyToDialog

MUIApplyToDialog, hWin:*MUIWND*, bDropShadow:BOOL, bClipping:BOOL

Applies the ModernUI style to a dialog to make it a captionless, borderless form. User can manually change a form in a resource editor to have the following style flags: WS_POPUP or WS_VISIBLE and optionally with DS_CENTER, DS_CENTERMOUSE, WS_CLIPCHILDREN, WS_CLIPSIBLINGS, WS_MINIMIZE, WS_MAXIMIZE

Parameters

- [in] **hWin** - handle to window to apply the ModernUI look
- [in] **bDropShadow** - show drop shadow on dialog/window TRUE or FALSE
- [in] **bClipping** - enable clipping on dialog/window TRUE or FALSE

Return

None

Example

```
Invoke MUIApplyToDialog, TRUE, FALSE
```

2.10.2 MUICenterWindow

MUICenterWindow, hWndChild:*MUIWND*, hWndParent:*MUIWND*

Center a child window **hWndChild** into parent window **hWndParent** (or desktop if **hWndParent** is NULL). Parent doesn't need to be the owner.

Parameters

- [in] **hWndChild** - handle to child window to center
- [in] **hWndParent** - parent window of child to center, relative to

Return

None

Example

```
Invoke GetParent, hWin
mov hParent, eax
Invoke MUICenterWindow, hWin, hParent
```

See Also

MUIGetParentRelativeWindowRect

2.10.3 MUIGetParentRelativeWindowRect

MUIGetParentRelativeWindowRect, hWin:*MUIWND*, lpRectControl:*LPRECT*

Get rectangle of a window/control relative to it's parent.

Parameters

- [in] **hWin** - handle to the window to map the RECT (relative to it's parent) passed as the **lpRectControl** variable
- [in/out] **lpRectControl** - pointer to a RECT variable used to pass and store the rectangle to be adjusted relative to the parent of **hWin**

Return

Returns TRUE if successful, or FALSE otherwise. **lpRectControl** on successful return will contain the newly mapped rectangle coordinates.

Example

```
LOCAL rect:RECT

Invoke GetClientRect, hWin, Addr rect
Invoke MUIGetParentRelativeWindowRect, hWin, Addr rect
```

See Also

MUICenterWindow

2.11 ModernUI Macros

Contents:

2.11.1 MUI_RGBCOLOR Macro

Creates a Red, Green, Blue COLORREF colour value used for **GDI** functions.

```
MUI_RGBCOLOR MACRO red:REQ, green:REQ, blue:REQ
    EXITM < red or green shl 8 or blue shl 16 >
ENDM
```

Parameters

- [in] **red** - 0–255 red color value
- [in] **green** - 0–255 green color value
- [in] **blue** - 0–255 blue color value

Return

Returns the RGB COLORREF value

Example

```
mov BackColor, MUI_RGBCOLOR(127, 200, 240)
```

See Also

RGB, MUI_ARGBCOLOR

2.11.2 MUI_ARGBCOLOR Macro

Creates an Alpha, Red, Green, Blue colour value used for **GDI+** functions.

```
MUI_ARGBCOLOR MACRO alpha, red, green, blue
    EXITM < alpha SHL 24 OR red SHL 16 OR green SHL 8 OR blue >
ENDM
```

Parameters

- [in] **alpha** - 0–255 alpha transparency value
- [in] **red** - 0–255 red color value
- [in] **green** - 0–255 green color value
- [in] **blue** - 0–255 blue color value

Return

Returns the ARGB value

Example

```
mov BackColor, MUI_ARGBCOLOR(255, 127, 200, 240)
```

See Also

AlphaRGB, MUI_RGBCOLOR

2.11.3 RGB Macro

Creates a Red, Green, Blue COLORREF colour value used for **GDI** functions.

```
RGB MACRO red:REQ, green:REQ, blue:REQ
    EXITM < red or green shl 8 or blue shl 16 >
ENDM
```

Parameters

- [in] **red** - 0–255 red color value
- [in] **green** - 0–255 green color value
- [in] **blue** - 0–255 blue color value

Return

Returns the RGB COLORREF value

Example

```
mov BackColor, RGB(127, 200, 240)
```

See Also

MUI_RGBCOLOR, MUI_ARGBCOLOR

2.11.4 AlphaRGB Macro

Creates an Alpha, Red, Green, Blue colour value used for **GDI+** functions.

```
AlphaRGB MACRO alpha, red, green, blue
    EXITM < alpha SHL 24 OR red SHL 16 OR green SHL 8 OR blue >
ENDM
```

Parameters

- [in] **alpha** - 0–255 alpha transparency value
- [in] **red** - 0–255 red color value
- [in] **green** - 0–255 green color value
- [in] **blue** - 0–255 blue color value

Return

Returns the ARGB value

Example

```
mov BackColor, AlphaRGB(255, 127, 200, 240)
```

See Also

MUI_ARGBCOLOR, MUI_RGBCOLOR

2.12 ModernUI Library Data Types

The following typedef or data types are used to help with documentation of ModernUI function parameters and return values, and with syncing ModernUI x86/x64 development by using one include file for both architectures: **ModernUI.inc**

2.12.1 ModernUI Data Types

These data types are specific to the ModernUI library:

Data type	x86 size	x64 size
<i>MUIWND</i>	DWORD	QWORD
<i>MUIPROPERTIES</i>	DWORD	QWORD
<i>MUIPROPERTY</i>	DWORD	QWORD
<i>MUIPROPERTYVALUE</i>	DWORD	QWORD
<i>MUIVALUE</i>	DWORD	QWORD
<i>LPMUIVALUE</i>	DWORD	QWORD
<i>MUIIT</i>	DWORD	QWORD
<i>MUIIL</i>	DWORD	QWORD
<i>MUIPFS</i>	DWORD	QWORD
<i>MUICOLORRGB</i>	DWORD	QWORD
<i>MUICOLORARGB</i>	DWORD	QWORD
<i>MUIIMAGE</i>	DWORD	QWORD
<i>LPMUIIMAGE</i>	DWORD	QWORD

2.12.2 GDI+ Data Types

These data types expand on or add to some of the GDI+ data types:

Data type	x86 size	x64 size
<i>GPGRAPHICS</i>	DWORD	QWORD
<i>LPGPGRAPHICS</i>	DWORD	QWORD
<i>GPRECT</i>	DWORD	QWORD
<i>LPGPRECT</i>	DWORD	QWORD
<i>LPGDIPRECT</i>	DWORD	QWORD
<i>GPIMAGE</i>	DWORD	QWORD
<i>LPGPIMAGE</i>	DWORD	QWORD

2.12.3 Common Data Types

These data types expand on or add to some of the common [Windows data types](#):

Data type	x86 size	x64 size
<i>LPRECT</i>	DWORD	QWORD
<i>LPHBITMAP</i>	DWORD	QWORD
<i>LP HDC</i>	DWORD	QWORD
<i>POINTER</i>	DWORD	QWORD
<i>RESID</i>	DWORD	QWORD

2.12.4 Data Types Description

MUIWND

Alias for [HWND](#) window handle for a ModernUI control handle, typically defined as `hWin`

MUIPROPERTIES

For `cbWndExtraOffset` parameter of *MUIAllocMemProperties* and *MUIFreeMemProperties* functions: `MUI_INTERNAL_PROPERTIES` or `MUI_EXTERNAL_PROPERTIES`

MUIPROPERTY

Enum for a specific ModernUI Control's property, using the `Property` parameter of *MUIGetIntProperty*, *MUISetIntProperty*, *MUIGetExtProperty* and *MUISetExtProperty* functions. See specific control for a list of its defined properties

MUIPROPERTYVALUE

Value of specific ModernUI Control's property, using the `PropertyValue` parameter of *MUISetIntProperty* or *MUISetExtProperty* functions

MUIVALUE

A value, a constant or typically an unsigned integer used in certain ModernUI function parameters

LPMUIVALUE

A pointer to a *MUIVALUE* value

MUIIT

Image type: `MUIIT_NONE`, `MUIIT_BMP`, `MUIIT_ICO` or `MUIIT_PNG`

MUIIL

Image location: MUIIL_CENTER, MUIIL_BOTTOMLEFT, MUIIL_BOTTOMRIGHT, MUIIL_TOPLEFT, MUIIL_TOPRIGHT, MUIIL_TOPCENTER, MUIIL_BOTTOMCENTER

MUIPFS

Paint frame style flags for the FrameStyle parameter of the *MUIGDIPaintFrame* function using a combination of flags: MUIPFS_NONE, MUIPFS_LEFT, MUIPFS_TOP, MUIPFS_BOTTOM, MUIPFS_RIGHT or MUIPFS_ALL

MUICOLORRGB

COLORREF color value using *MUI_RGBCOLOR* macro

MUICOLORARGB

ARGB color value using *MUI_ARGBCOLOR* macro

MUIIMAGE

A bitmap (*HBITMAP*), icon (*HICON*) or a GDI+ image (*GPIMAGE*)

LPMUIIMAGE

Pointer to a image (*MUIIMAGE*) handle

GPGRAPHICS

GDI+ graphics context

LPGPGRAPHICS

Pointer to a graphics context (*GPGRAPHICS*)

GPRECT

Alias for *GDIPRECT*, a rectangle using REAL4 (float) for coordinates

LPGPRECT

Pointer to *GPRECT*

LPGDIPRECT

Pointer to *GDIPRECT*

GPIMAGE

A GDI+ image

LPGPIMAGE

Pointer to GDI+ image (*GPIMAGE*)

LPRECT

Pointer to *RECT*

LPHBITMAP

Pointer to GDI bitmap (*HBITMAP*)

LPHDC

Pointer to *HDC*

POINTER

A pointer

RESID

A resource id value

2.13 ModernUI Library Structures

GDIPRECT Structure:

A rectangle structure that uses REAL4 (float) for coordinates instead of integers. This structure is used by certain **GDI+** functions that require float values.

GDIPRECT	STRUCT
left	REAL4 ?
top	REAL4 ?
right	REAL4 ?
bottom	REAL4 ?
GDIPRECT	ENDS

2.13.1 GDIPRECT Members

left

Specifies the x-coordinate of the upper-left corner of the rectangle.

top

Specifies the y-coordinate of the upper-left corner of the rectangle.

right

Specifies the x-coordinate of the lower-right corner of the rectangle.

bottom

Specifies the y-coordinate of the lower-right corner of the rectangle.

The functions, messages, structures and properties of the following ModernUI controls are documented in their own sections.

Contents:

3.1 ModernUI_Button



The ModernUI_Button is a button control like the standard win32 button control, except it provides ease of use and more customizable features, like color of text, background, border, accent and colors for when the mouse moves over the control, or the control is in a selected state.

Additionally images: icon, bitmap or png types, can be assigned to the ModernUI_Button control via the properties or by using one of the provided ModernUI_Button functions.

3.1.1 ModernUI_Button Functions

MUIButtonCreate

MUIButtonCreate, hWndParent:*MUIWND*, lpzText:LPSTR, X:*MUIVALUE*, Y:*MUIVALUE*, nWidth:*MUIVALUE*, nHeight:*MUIVALUE*, ResourceID:*RESID*, Style:*MUIVALUE*

Creates a new ModernUI_Button control.

Parameters

- [in] **hWndParent** - parent window of control
- [in] **lpzText** - text to display

- [in] **X** - x coord of control
- [in] **Y** - y coord of control
- [in] **nWidth** - width of control
- [in] **nHeight** - height of control
- [in] **ResourceID** - resource id of control
- [in] **Style** - can be combination of style flags, see *ModernUI_Button Style Flags* for details

Return

Returns handle to newly created ModernUI_Button control (MUIWND) if successful, or NULL otherwise

ModernUI_Button Style Flags

- MUIBS_LEFT - Align text to the left of the button
- MUIBS_BOTTOM - Place image at the top, and text below
- MUIBS_CENTER - Align text centerally.
- MUIBS_AUTOSTATE - Automatically toggle between TRUE/FALSE state when clicked. TRUE = Selected.
- MUIBS_PUSHBUTTON - Simulate button movement down slightly when mouse click and movement up again when mouse is released.
- MUIBS_HAND - Show a hand instead of an arrow when mouse moves over button.
- MUIBS_KEEPIMAGES - Dont delete image handles when control is destroyed. Essential if image handles are used in multiple controls.
- MUIBS_DROPDOWN - Show dropdown arrow right side of control
- MUIBS_NOFOCUSRECT - Dont show focus rect, just use change border to @ButtonBorderColorAlt when setfocus.
- MUIBS_THEME - Use default windows theme colors and react to WM_THEMECHANGED

Example

```
Invoke MUIButtonCreate, hWin, Addr szButtonText, 10, 10, 150, 30, IDC_BTN1, MUIBS_
↪PUSHBUTTON or MUIBS_HAND
```

See Also

MUIButtonRegister, MUIButtonGetProperty, MUIButtonSetProperty

MUIButtonGetProperty

MUIButtonGetProperty, hWin:*MUIWND*, Property:*MUIPROPERTY*

Get the value of a property in a ModernUI_Button control. See *ModernUI_Button Properties* for details on the properties available

Parameters

- [in] **hWin** - handle to the ModernUI_Button control
- [in] **Property** - the property to get. See *ModernUI_Button Properties* for details on the properties available

Return

Returns the value of the property or NULL otherwise

Example

```
LOCAL dwTextColor:DWORD

Invoke MUIButtonGetProperty, hTxt1, @ButtonTextColor
mov dwTextColor, eax
```

See Also

MUIButtonSetProperty, ModernUI_Button Properties

MUIButtonGetState**MUIButtonGetState****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘, :ref:‘
```

MUIButtonLoadImages**MUIButtonLoadImages****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘, :ref:‘
```

MUIButtonNoteSetFont

MUIButtonNoteSetFont

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIButtonNoteSetText

MUIButtonNoteSetText

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIButtonNotify

MUIButtonNotify

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIButtonNotifyLoadImage

MUIButtonNotifyLoadImage

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIButtonNotifySetFont

MUIButtonNotifySetFont

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIButtonNotifySetImage

MUIButtonNotifySetImage

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

****See Also****

:ref:‘, :ref:‘

MUIButtonNotifySetText

MUIButtonNotifySetText

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

****See Also****

:ref:‘, :ref:‘

MUIButtonRegister

MUIButtonRegister

Registers the ModernUI_Button class, which allows the control to be created via dialog resources or via [CreateWindowEx](#) and specifying the classname ‘ModernUI_Button’. If using the RadASM custom class control instead of a the ModernUI RadASM Design Time Controls specify ‘ModernUI_Button’ as the classname

Parameters

None

Return

None

Example

Invoke MUIButtonRegister

See Also

MUIButtonCreate, MUIButtonGetProperty, MUIButtonSetProperty

MUIButtonSetAllProperties

MUIButtonSetAllProperties

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘, :ref:‘
```

MUIButtonSetImages**MUIButtonSetImages****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘, :ref:‘
```

MUIButtonSetProperty

MUIButtonSetProperty, hWin:[MUIWND](#), Property:[MUIPROPERTY](#), PropertyValue:[MUIPROPERTYVALUE](#)

Sets the value of a property in a ModernUI_Button control. See [ModernUI_Button Properties](#) for details on the properties available

Parameters

- [in] **hWin** - handle to the ModernUI_Button control
- [in] **Property** - the property to set. See [ModernUI_Button Properties](#) for details on the properties available
- [in] **PropertyValue** - the value to set the property to

Return

Returns the previously set value for the property, or NULL otherwise

Example

```
Invoke MUIButtonSetProperty, hTxt1, @ButtonTextColor, MUI_RGBCOLOR(48,48,48)
```

See Also

[MUIButtonGetProperty](#), [ModernUI_Button Properties](#)

MUIButtonSetState

MUIButtonSetState

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

See Also

:ref:‘ , :ref: ‘

Function	Description
<i>MUIButtonCreate</i>	Creates a new ModernUI_Button control
<i>MUIButtonGetProperty</i>	Gets the value of a property
<i>MUIButtonGetState</i>	Gets the current state: selected, unselected
<i>MUIButtonLoadImages</i>	Load images from resources and assign to the control
<i>MUIButtonNoteSetFont</i>	<i>Deprecated</i>
<i>MUIButtonNoteSetText</i>	<i>Deprecated</i>
<i>MUIButtonNotify</i>	<i>Deprecated</i>
<i>MUIButtonNotifyLoadImage</i>	<i>Deprecated</i>
<i>MUIButtonNotifySetFont</i>	<i>Deprecated</i>
<i>MUIButtonNotifySetImage</i>	<i>Deprecated</i>
<i>MUIButtonNotifySetText</i>	<i>Deprecated</i>
<i>MUIButtonRegister</i>	Registers a window class for the ModernUI_Button
<i>MUIButtonSetAllProperties</i>	Sets all properties in one function using a structure
<i>MUIButtonSetImages</i>	Assigns images (previously loaded) to the control
<i>MUIButtonSetProperty</i>	Sets the value of a property
<i>MUIButtonSetState</i>	Sets the state: selected, unselected

3.1.2 ModernUI_Button Messages

Message	Description
MUIBM_GETSTATE	Gets the current state: selected, unselected
MUIBM_SETSTATE	Sets the state: selected, unselected
MUIBM_NOTIFYSETTEXT	<i>Deprecated</i>
MUIBM_NOTIFYSETIMAGE	<i>Deprecated</i>
MUIBM_NOTIFYLOADIMAGE	<i>Deprecated</i>
MUIBM_NOTIFYSETFONT	<i>Deprecated</i>
MUIBM_NOTIFY	<i>Deprecated</i>
MUIBM_NOTESETTEXT	<i>Deprecated</i>
MUIBM_NOTESETFONT	<i>Deprecated</i>

3.1.3 ModernUI_Button Properties

Property	Type	Notes
ButtonTextFont	HFONT	
ButtonTextColor	COLORREF	
ButtonTextColorAlt	COLORREF	
ButtonTextColorSel	COLORREF	
ButtonTextColorSelAlt	COLORREF	
ButtonTextColorDisabled	COLORREF	
ButtonBackColor	COLORREF	-1 = transparent
ButtonBackColorAlt	COLORREF	
ButtonBackColorSel	COLORREF	
ButtonBackColorSelAlt	COLORREF	
ButtonBackColorDisabled	COLORREF	
ButtonBorderColor	COLORREF	-1 = transparent
ButtonBorderColorAlt	COLORREF	
ButtonBorderColorSel	COLORREF	
ButtonBorderColorSelAlt	COLORREF	
ButtonBorderColorDisabled	COLORREF	
ButtonBorderStyle	DWORD	Button Border Styles
ButtonAccentColor	COLORREF	-1 = transparent
ButtonAccentColorAlt	COLORREF	
ButtonAccentColorSel	COLORREF	
ButtonAccentColorSelAlt	COLORREF	
ButtonAccentStyle	DWORD	Button Accent Styles
ButtonAccentStyleAlt	DWORD	Button Accent Styles
ButtonAccentStyleSel	DWORD	Button Accent Styles
ButtonAccentStyleSelAlt	DWORD	Button Accent Styles
ButtonImageType	DWORD	Button Image Types
ButtonImage	hImage	
ButtonImageAlt	hImage	
ButtonImageSel	hImage	
ButtonImageSelAlt	hImage	
ButtonImageDisabled	hImage	
ButtonRightImage	hImage	Right side image
ButtonRightImageAlt	hImage	Right side image
ButtonRightImageSel	hImage	Right side image
ButtonRightImageSelAlt	hImage	Right side image
ButtonRightImageDisabled	hImage	Right side image
ButtonNotifyTextFont	HFONT	
ButtonNotifyTextColor	COLORREF	
ButtonNotifyBackColor	COLORREF	
ButtonNotifyRound	DWORD	dwPixels - Roundrect x,y value
ButtonNotifyImageType	DWORD	Button Image Types
ButtonNotifyImage	hImage	
ButtonNoteTextFont	HFONT	
ButtonNoteTextColor	COLORREF	
ButtonNoteTextColorDisabled	COLORREF	
ButtonPaddingLeftIndent	DWORD	dwPixels - No of pixels to indent images + text (or just text
ButtonPaddingGeneral	DWORD	dwPixels - No of pixels of padding to apply (Default 4px).

Continued on next page

Table 1 – continued from previous page

ButtonPaddingStyle	Button	Padding Style
ButtonPaddingTextImage	DWORD	dwPixels - No of pixels between left images and text. (Default 8px)
ButtonDllInstance	DWORD	Set to hInstance of dll if loading images
ButtonParam	DWORD	Custom user data

3.2 ModernUI_CaptionBar



The ModernUI_CaptionBar is a control set at the top of your window / dialog, that comprises of a rectangle containing the caption text or title of the window / dialog, along with one or more system buttons. System buttons are typically minimize, maximize, restore and close.

The ModernUI_CaptionBar control is automatically aligned to the top of the window dialog, and spans the width of it. The control will automatically adjust the width when the window / dialog is resized, via the system buttons (min/max/restore), double clicking the caption bar itself (toggles between maximized and restored) or programmatically via ShowWindow or other win32 api calls.

3.2.1 ModernUI_CaptionBar Functions

MUICapButtonGetProperty

MUICapButtonGetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref: ' , :ref: '
```

MUICapButtonSetProperty

MUICapButtonSetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUICaptionBarAddButton**MUICaptionBarAddButton****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUICaptionBarAddButtonEx**MUICaptionBarAddButtonEx****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUICaptionBarCreate**MUICaptionBarCreate****Parameters**

- [in] 1

- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘ , :ref: ‘

MUICaptionBarGetProperty

MUICaptionBarGetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘ , :ref: ‘

MUICaptionBarLoadBackImage

MUICaptionBarLoadBackImage

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘ , :ref: ‘

MUICaptionBarLoadIcons

MUICaptionBarLoadIcons

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUICaptionBarLoadIconsDll

MUICaptionBarLoadIconsDll

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUICaptionBarRegister

MUICaptionBarRegister

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUICaptionBarSetProperty

MUICaptionBarSetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

3.2.2 ModernUI_CaptionBar Messages

3.2.3 ModernUI_CaptionBar Properties

3.3 ModernUI_Checkbox



The ModernUI_Checkbox is a check or radio control like the standard win32 controls of that type, except it provides ease of use and more customizable features, like color of text, background, border and colors for when the mouse moves over the control, or the control is in a selected state.

Additionally custom check or radio images can be assigned to the ModernUI_Checkbox control via the properties or by using one of the provided ModernUI_Checkbox functions.

3.3.1 ModernUI_Checkbox Functions

MUICheckboxCreate

MUICheckboxCreate

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUICheckboxGetProperty**MUICheckboxGetProperty****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUICheckboxGetState**MUICheckboxGetState****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUICheckboxLoadImages**MUICheckboxLoadImages****Parameters**

- [in] 1

- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘ , :ref: ‘

MUICheckboxRegister

MUICheckboxRegister

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘ , :ref: ‘

MUICheckboxSetImages

MUICheckboxSetImages

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘ , :ref: ‘

MUICheckboxSetProperty

MUICheckboxSetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref: ' , :ref: '
```

MUICheckboxSetState

MUICheckboxSetState

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref: ' , :ref: '
```

3.3.2 ModernUI_Checkbox Messages

3.3.3 ModernUI_Checkbox Properties

3.4 ModernUI_ProgressBar



The ModernUI_ProgressBar is a progress bar control like the standard win32 progress bar control, except it provides ease of use and more customizable features, like color of the progress fill, the progress unfilled area and border.

3.4.1 ModernUI_ProgressBar Functions

MUIProgressBarCreate

MUIProgressBarCreate, hWndParent:*MUIWND*, X:*MUIVALUE*, Y:*MUIVALUE*, nWidth:*MUIVALUE*, nHeight:*MUIVALUE*, ResourceID:*RESID*, Style:*MUIVALUE*

Creates a new ModernUI_ProgressBar control

Parameters

- [in] **hWndParent** - parent window of control
- [in] **X** - x coordinate of the control
- [in] **Y** - y coordinate of the control
- [in] **nWidth** - width of the control
- [in] **nHeight** - height of the control
- [in] **ResourceID** - resource id of the control
- [in] **Style** - 0

Return

Returns handle to newly created ModernUI_SmartPanel control (MUIWND) if successful, or NULL otherwise

Example

```
Invoke MUIProgressBarCreate, hWin, 100, 100, 300, 14, IDC_PBAR, 0
```

See Also

MUIProgressBarRegister, MUIProgressBarGetProperty, MUIProgressBarSetProperty

MUIProgressBarGetPercent

MUIProgressBarGetPercent, hWin:*MUIWND*

Gets the current percent value of the progressbar

Parameters

- [in] **hWin** - handle to the ModernUI_ProgressBar control

Return

Returns the current percentage value

Example

```
Invoke MUIProgressBarGetPercent, hProgressBar
```

See Also

MUIProgressBarSetPercent

MUIProgressBarGetProperty

MUIProgressBarGetProperty, hWin:[MUIWND](#), Property:[MUIPROPERTY](#)

Get the value of a property in a ModernUI_ProgressBar control. See [ModernUI_ProgressBar Properties](#) for details on the properties available

Parameters

- [in] **hWin** - handle to the ModernUI_ProgressBar control
- [in] **Property** - the property to get. See [ModernUI_ProgressBar Properties](#) for details on the properties available

Return

Returns the value of the property or NULL otherwise

Example

```
LOCAL ProgressColor:DWORD

Invoke MUIProgressBarGetProperty, hProgressBar, @ProgressBarProgressColor
mov ProgressColor, eax
```

See Also

[MUIProgressBarSetProperty](#), [ModernUI_ProgressBar Properties](#)

MUIProgressBarRegister

MUIProgressBarRegister

Registers the ModernUI_ProgressBar class, which allows the control to be created via dialog resources or via [CreateWindowEx](#) and specifying the classname 'ModernUI_ProgressBar'. If using the RadASM custom class control instead of a the ModernUI RadASM Design Time Controls specify 'ModernUI_ProgressBar' as the classname

Parameters

None

Return

None

Example

```
Invoke MUIProgressBarRegister
```

See Also

[MUIProgressBarCreate](#), [MUIProgressBarGetProperty](#), [MUIProgressBarSetProperty](#)

MUIProgressBarSetMinMax

MUIProgressBarSetMinMax

Parameters

- [in] **1**
- [in] **2**
- [in] **3**

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIProgressBarSetPercent

MUIProgressBarSetPercent, hWin:*MUIWND*, Percent:*MUIVALUE*

Sets the current percent value of the progressbar

Parameters

- [in] **hWin** - handle to the ModernUI_ProgressBar control
- [in] **Percent** - value to set percentage of the progress bar to

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUIProgressBarSetPercent, hProgressBar, 50
```

See Also

MUIProgressBarGetPercent, *MUIProgressBarStep*

MUIProgressBarSetProperty

MUIProgressBarSetProperty, hWin:*MUIWND*, Property:*MUIPROPERTY*, PropertyValue:*MUIPROPERTYVALUE*

Sets the value of a property in a ModernUI_ProgressBar control. See *ModernUI_ProgressBar Properties* for details on the properties available

Parameters

- [in] **hWin** - handle to the ModernUI_ProgressBar control
- [in] **Property** - the property to set. See *ModernUI_ProgressBar Properties* for details on the properties available
- [in] **PropertyValue** - the value to set the property to

Return

Returns the previously set value for the property, or NULL otherwise

Example

```
Invoke MUIProgressBarSetProperty, hProgressBar, @ProgressBarProgressColor, MUI_
↪RGBCOLOR(67,104,210)
```

See Also

MUIProgressBarGetProperty, *ModernUI_ProgressBar Properties*

MUIProgressBarStep

MUIProgressBarStep, hWin:*MUIWND*

Incrementally moves the progressbar

Parameters

- [in] **hWin** - handle to the ModernUI_ProgressBar control

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUIProgressBarStep, hProgressBar
```

See Also

MUIProgressBarSetPercent

Function	Description
<i>MUIProgressBarCreate</i>	Creates a new ModernUI_ProgressBar control
<i>MUIProgressBarGetPercent</i>	Gets the current percent value of the progressbar
<i>MUIProgressBarGetProperty</i>	Gets the value of a property
<i>MUIProgressBarRegister</i>	Registers a window class for the ModernUI_ProgressBar
<i>MUIProgressBarSetMinMax</i>	Sets the minimum and maximum range of progressbar
<i>MUIProgressBarSetPercent</i>	Sets the current percent value of the progressbar
<i>MUIProgressBarSetProperty</i>	Sets the value for a property
<i>MUIProgressBarStep</i>	Incrementally moves the progressbar

3.4.2 ModernUI_ProgressBar Messages

MUIPBM_SETPERCENT

MUIPBM_SETPERCENT EQU WM_USER + 1749

A custom windows message for the control that sets the current percent value of the progressbar

Parameters

- **wParam** - value to set percentage of the progress bar to
- **lParam** - NULL

Return

None

Example

```
Invoke SendMessage, hProgressBar, MUIPBM_SETPERCENT, 50, NULL
```

See Also

MUIPBM_STEP

MUIPBM_STEP

MUIPBM_STEP EQU WM_USER + 1750

A custom windows message for the control that incrementally moves the progressbar

Parameters

- **wParam** - NULL
- **lParam** - NULL

Return

None

Example

```
Invoke SendMessage, hProgressBar, MUIPBM_STEP, NULL, NULL
```

See Also

MUIPBM_SETPERCENT

Message	Description
<i>MUIPBM_SETPERCENT</i>	Sets the percentage of the progressbar
<i>MUIPBM_STEP</i>	Incrementally moves the progressbar one step

3.4.3 ModernUI_ProgressBar Properties

Property	Type
<i>@ProgressBarTextColor</i>	MUICOLORRGB
<i>@ProgressBarTextFont</i>	HFONT
<i>@ProgressBarBackColor</i>	MUICOLORRGB
<i>@ProgressBarProgressColor</i>	MUICOLORRGB
<i>@ProgressBarBorderColor</i>	MUICOLORRGB
<i>@ProgressBarPercent</i>	MUIVALUE
<i>@ProgressBarMin</i>	MUIVALUE
<i>@ProgressBarMax</i>	MUIVALUE
<i>@ProgressBarStep</i>	MUIVALUE
<i>@ProgressBarPulse</i>	BOOL
<i>@ProgressBarPulseTime</i>	MUIVALUE
<i>@ProgressBarTextType</i>	MUIVALUE
<i>@ProgressBarSetTextPos</i>	MUIVALUE

ModernUI_ProgressBar Property Descriptions

ProgressBarTextColor

Color of percentage text (*MUICOLORRGB*) of the ModernUI_ProgressBar control.

ProgressBarTextFont

Font (HFONT) used for the progress bar text

ProgressBarBackColor

Background color (*MUICOLORRGB*) of the ModernUI_ProgressBar control.

ProgressBarProgressColor

Progress bar percent filled color (*MUICOLORRGB*) of the ModernUI_ProgressBar control.

ProgressBarBorderColor

Border color (*MUICOLORRGB*) of the ModernUI_ProgressBar control.

ProgressBarPercent

Current progress bar percentage value

ProgressBarMin

Minimum range value - *currently not implemented*

ProgressBarMax

Maximum range value - *currently not implemented*

ProgressBarStep

Value to increment the progress bar when calling *MUIProgressBarStep* or *MUIPBM_STEP*. Defaults to 1 - *currently not implemented*

ProgressBarPulse

Enable pulse glow effect to show progress bar is still active. TRUE to enable, FALSE to disable. Default is TRUE

ProgressBarPulseTime

Time in milliseconds between pulse effect is shown. Defaults to 5 seconds (5000ms)

ProgressBarTextType

Type of percentage text to display, can be one of the following values:

- MUIPBTT_NONE- no percentage text in progress bar (default)
- MUIPBTT_CENTRE - percentage text in center of progress bar
- MUIPBTT_FOLLOW - percentage text follows progress as it draws

ProgressBarSetTextPos

Position of other text, can be one of the following values: 0 = prepend WM_SETTEXT text, 1 = append WM_SETTEXT text *currently not implemented*

3.5 ModernUI_ProgressDots



The ModernUI_ProgressDots is a control similar to a progress bar, but features small dots that travel horizontally across the screen, and the dots vary in speed at which they travel depending on their location. They can be seen to slow down in the middle third of the screen, and speed up at the beginning and end thirds. Typically this type of control is used to indicate progress of an indeterminate length of time.

3.5.1 ModernUI_ProgressDots Functions

MUIProgressDotsAnimateStart

MUIProgressDotsAnimateStart

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIProgressDotsAnimateStop

MUIProgressDotsAnimateStop

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIProgressDotsCreate

MUIProgressDotsCreate

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref:‘
```

MUIProgressDotsGetProperty

MUIProgressDotsGetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref:‘
```

MUIProgressDotsRegister

MUIProgressDotsRegister

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref:‘
```

MUIProgressDotsSetProperty

MUIProgressDotsSetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘, :ref:‘
```

3.5.2 ModernUI_ProgressDots Messages

3.5.3 ModernUI_ProgressDots Properties

3.6 ModernUI_Region



The ModernUI_RegionButton is a button control like the standard win32 button control, except it is an irregular shape, defined by polygons or premade regions. It provides ease of use and more customizable features, like color of background, border, and colors for when the mouse moves over the control, or the control is in a selected state, and can use a bitmap brush to paint a collection of ModernUI_RegionButton controls, for example for maps: country, county and/or state regions.

3.6.1 ModernUI_Region Functions

MUIRegionButtonCustomStates

MUIRegionButtonCustomStates

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘, :ref:‘
```

MUIRegionButtonCreate

MUIRegionButtonCreate

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIRegionButtonGetCustomState**MUIRegionButtonGetCustomState****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIRegionButtonGetProperty**MUIRegionButtonGetProperty****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIRegionButtonGetState

MUIRegionButtonGetState

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIRegionButtonRegister

MUIRegionButtonRegister

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIRegionButtonSetBitmap

MUIRegionButtonSetBitmap

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIRegionButtonSetBrush

MUIRegionButtonSetBrush

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIRegionButtonSetCustomState

MUIRegionButtonSetCustomState

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUIRegionButtonSetProperty

MUIRegionButtonSetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

****See Also****

:ref:‘, :ref:‘

MUIRegionButtonSetRegion

MUIRegionButtonSetRegion

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

****See Also****

:ref:‘, :ref:‘

MUIRegionButtonSetRegionBitmap

MUIRegionButtonSetRegionBitmap

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

****See Also****

:ref:‘, :ref:‘

MUIRegionButtonSetRegionPoly

MUIRegionButtonSetRegionPoly

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘, :ref:‘
```

MUIRegionButtonSetState**MUIRegionButtonSetState****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘, :ref:‘
```

3.6.2 ModernUI_Region Messages**3.6.3 ModernUI_Region Properties****3.7 ModernUI_SmartPanel**

The ModernUI_SmartPanel is an invisible control - it is only shown during design time (if using the RadASM design-time dll for the ModernUI_SmartPanel). Its purpose is to host other dialog panels, and facilitate moving between dialog panels seamlessly. It can also provide a slide effect when changing from one panel to another.

The ModernUI_SmartPanel control can also control the painting of the background for the dialog panels that are registered with it. Additionally it will adjust each panel's style at registration so that it is flat, borderless and without a caption, and set the DS_CONTROL flag for its style. If using IsDialogMessage in your message loop, the ModernUI_SmartPanel control can store the handle of the currently used dialog panel, in a variable that can be used with the IsDialogMessage during the event loop, so that you can provide tabbing between controls of the hosted dialog panel's controls.

3.7.1 ModernUI_SmartPanel Functions

MUISmartPanelCreate

MUISmartPanelCreate, hWndParent:*MUIWND*, X:*MUIVALUE*, Y:*MUIVALUE*, nWidth:*MUIVALUE*, nHeight:*MUIVALUE*, ResourceID:*RESID*, Style:*MUIVALUE*

Creates a new ModernUI_SmartPanel control

Parameters

- [in] **hWndParent** - parent window of control
- [in] **X** - x coordinate of the control
- [in] **Y** - y coordinate of the control
- [in] **nWidth** - width of the control
- [in] **nHeight** - height of the control
- [in] **ResourceID** - resource id of the control
- [in] **Style** - can be combination of style flags, see *ModernUI_SmartPanel Style Flags* for details

Return

Returns handle to newly created ModernUI_SmartPanel control (MUIWND) if successful, or NULL otherwise

ModernUI_SmartPanel Style Flags

- MUISPS_NORMAL - no slide animation
- MUISPS_NOSLIDE - no slide animation
- MUISPS_SLIDEPANELS_SLOW - slow speed slide animation
- MUISPS_SLIDEPANELS_NORMAL - normal speed slide animation
- MUISPS_SLIDEPANELS - normal speed slide animation
- MUISPS_SLIDEPANELS_FAST - fast speed slide animation
- MUISPS_SLIDEPANELS_VFAST - very fast speed slide animation
- MUISPS_SLIDEPANELS_INSTANT - no slide animation
- MUISPS_SPS_WRAPAROUND - for next/prev and showcase, if at end, moves to the right and starts again, otherwise if not specified, at last panel, scrolls left all the way back to start showing all panels along the way.
- MUISPS_SPS_SKIPBETWEEN - skips any in between panels, just moves from one to another.
- MUISPS_DESIGN_INFO - only used at design time to show text, which can be toggled off by user

Example

```
Invoke MUISmartPanelCreate, hWin, 10, 10, 1024, 800, IDC_SMARTPANEL, MUISPS_NOSLIDE
```

See Also

MUISmartPanelRegister, *MUISmartPanelGetProperty*, *MUISmartPanelSetProperty*, *MUISmartPanelRegisterPanel*

MUISmartPanelCurrentPanelIndex

MUISmartPanelCurrentPanelIndex, hWin:*MUIWND*

Gets the current panel that is active in the ModernUI_SmartPanel control and returns the panel index of this dialog

Parameters

- [in] **hWin** - handle to the ModernUI_SmartPanel control

Return

Returns current panel index or -1 if there is no panels registered

Example

```
Invoke MUISmartPanelCurrentPanelIndex, hSP
```

See Also

MUISmartPanelGetCurrentPanel, MUISmartPanelSetCurrentPanel

MUISmartPanelGetCurrentPanel

MUISmartPanelGetCurrentPanel, hWin:*MUIWND*

Gets the current panel that is active in the ModernUI_SmartPanel control and returns the handle to this dialog

Parameters

- [in] **hWin** - handle to the ModernUI_SmartPanel control

Return

Returns handle to current panel (dialog window) (MUIWND)

Example

```
Invoke MUISmartPanelGetCurrentPanel, hSP
```

See Also

MUISmartPanelSetCurrentPanel, MUISmartPanelCurrentPanelIndex

MUISmartPanelGetPanelParam

MUISmartPanelGetPanelParam

Parameters

- [in] **1**
- [in] **2**
- [in] **3**

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref: ‘, :ref: ‘

MUISmartPanelGetProperty

MUISmartPanelGetProperty, hWin:*MUIWND*, Property:*MUIPROPERTY*

Get the value of a property in a ModernUI_SmartPanel control. See *ModernUI_SmartPanel Properties* for details on the properties available

Parameters

- [in] **hWin** - handle to the ModernUI_SmartPanel control
- [in] **Property** - the property to get. See *ModernUI_SmartPanel Properties* for details on the properties available

Return

Returns the value of the property or NULL otherwise

Example

```
LOCAL BackColor:DWORD

Invoke MUISmartPanelGetProperty, hSP, @SmartPanelBackColor
mov BackColor, eax
```

See Also

MUISmartPanelSetProperty, *ModernUI_SmartPanel Properties*

MUISmartPanelNextPanel

MUISmartPanelNextPanel, hWin:*MUIWND*, bNotify:BOOL

Sets the active current panel to the next panel. If style `MUISPS_SPS_WRAPAROUND` is specified at creation, the panel will move and wrap around from the last panel to the first panel if it is currently at the last panel, instead of doing nothing. If panels are set to slide (animate) with any of the styles `MUISPS_SLIDEPANELS_SLOW`, `MUISPS_SLIDEPANELS_NORMAL`, `MUISPS_SLIDEPANELS_FAST`, `MUISPS_SLIDEPANELS_VFAST` they will slide from left to right. **bNotify** is an optional parameter that if TRUE will send a `WM_NOTIFY` message to the parent of the ModernUI_SmartPanel control with a code of `MUISPN_SELCHANGED` using a `NM_MUISMARTPANEL` structure

Parameters

- [in] **hWin** - handle to the ModernUI_SmartPanel control
- [in] **bNotify** - post notification that current panel has changed

Return

None

Example

```
Invoke MUISmartPanelNextPanel, hSP, TRUE
```

See Also

MUISmartPanelPrevPanel, *MUISmartPanelSetCurrentPanel*, *ModernUI_SmartPanel Style Flags*

MUISmartPanelNotifyCallback

MUISmartPanelNotifyCallback

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUISmartPanelPrevPanel

MUISmartPanelPrevPanel, hWin:*MUIWND*, bNotify:BOOL

Sets the active current panel to the previous panel. If style MUISPS_SPS_WRAPAROUND is specified at creation, the panel will move and wrap around from the first panel to the last panel if it is currently at the first panel, instead of doing nothing. If panels are set to slide (animate) with any of the styles MUISPS_SLIDEPANELS_SLOW, MUISPS_SLIDEPANELS_NORMAL, MUISPS_SLIDEPANELS, MUISPS_SLIDEPANELS_FAST, MUISPS_SLIDEPANELS_VFAST they will slide from right to left. **bNotify** is an optional parameter that if TRUE will send a [WM_NOTIFY](#) message to the parent of the ModernUI_SmartPanel control with a code of MUISPN_SELCHANGED using a NM_MUISMARTPANEL structure

Parameters

- [in] **hWin** - handle to the ModernUI_SmartPanel control
- [in] **bNotify** - post notification that current panel has changed

Return

None

Example

```
Invoke MUISmartPanelPrevPanel, hSP, TRUE
```

See Also

MUISmartPanelNextPanel, MUISmartPanelSetCurrentPanel, ModernUI_SmartPanel Style Flags

MUISmartPanelRegister

MUISmartPanelRegister

Registers the ModernUI_SmartPanel class, which allows the control to be created via dialog resources or via [CreateWindowEx](#) and specifying the classname ‘ModernUI_SmartPanel’. If using the RadASM custom class control instead of a the ModernUI RadASM Design Time Controls specify ‘ModernUI_SmartPanel’ as the classname

Parameters

None

Return

None

Example

```
Invoke MUISmartPanelRegister
```

See Also

MUISmartPanelCreate, *MUISmartPanelGetProperty*, *MUISmartPanelSetProperty*, *MUISmartPanelRegisterPanel*

MUISmartPanelRegisterPanel

MUISmartPanelRegisterPanel, hWin:*MUIWND*, ResIdPanelDlg:*RESID*, lpPanelProc:*POINTER*

Registers a dialog panel to be used with the ModernUI_SmartPanel control. The dialogs are created by the ModernUI_SmartPanel control and are hidden until they are set to be active, by calls to *MUISmartPanelSetCurrentPanel* or *MUISmartPanelNextPanel* or *MUISmartPanelPrevPanel*.

Parameters

- [in] **hWin** - handle to the ModernUI_SmartPanel control
- [in] **ResIdPanelDlg** - resource id of dialog to register
- [in] **lpPanelProc** - address of dialog's main procedure

Return

Returns handle to newly created and registered panel MUIWND if successful, or NULL otherwise

Example

```
InfoDialogProc PROTO :HWND, :UINT, :WPARAM, :LPARAM
TestDialogProc PROTO :HWND, :UINT, :WPARAM, :LPARAM

.const
IDD_INFO_DIALOG EQU 2000 ; resource id assigned to dialog
IDD_TEST_DIALOG EQU 3000 ; resource id assigned to dialog
```

```
InfoDialogProc PROC hWin:HWND, uMsg:UINT, wParam:WPARAM, lParam:LPARAM
    mov eax, uMsg
    .IF eax == WM_INITDIALOG

    .ELSEIF eax == WM_COMMAND

    .ELSEIF eax==WM_CLOSE
        Invoke DestroyWindow, hWin

    .ELSE
        mov eax, FALSE
        ret
    .ENDIF
    mov eax, TRUE
    ret
InfoDialogProc ENDP

TestDialogProc PROC hWin:HWND, uMsg:UINT, wParam:WPARAM, lParam:LPARAM
```

(continues on next page)

(continued from previous page)

```

mov eax, uMsg
.IF eax == WM_INITDIALOG

.ELSEIF eax == WM_COMMAND

.ELSEIF eax==WM_CLOSE
    Invoke DestroyWindow, hWin

.ELSE
    mov eax, FALSE
    ret
.ENDIF
mov eax, TRUE
ret
TestDialogProc ENDP

```

```

; Register dialog panels with our Modern_SmartPanel control
Invoke MUISmartPanelRegisterPanel, hSP, IDD_INFO_DIALOG, Addr InfoDialogProc
Invoke MUISmartPanelRegisterPanel, hSP, IDD_TEST_DIALOG, Addr TestDialogProc

Invoke MUISmartPanelSetCurrentPanel, hSP, 0 ; set to first registered panel

```

See Also

MUISmartPanelSetIsDlgMsgVar, MUISmartPanelNextPanel, MUISmartPanelPrevPanel, MUISmartPanelSetCurrentPanel

MUISmartPanelSetCurrentPanel

MUISmartPanelSetCurrentPanel, hWin:*MUIWND*, PanelIndex:*MUIVALUE*, bNotify:BOOL

Sets the active panel to show, as specified via the panel index parameter PanelIndex. If panels are set to slide (animate) with any of the styles MUISPS_SLIDEPANELS_SLOW, MUISPS_SLIDEPANELS_NORMAL, MUISPS_SLIDEPANELS, MUISPS_SLIDEPANELS_FAST, MUISPS_SLIDEPANELS_VFAST they will slide from left to right. **bNotify** is an optional parameter that if TRUE will send a **WM_NOTIFY** message to the parent of the ModernUI_SmartPanel control with a code of MUISPN_SELCHANGED using a NM_MUISMARTPANEL structure

Parameters

- [in] **hWin** - handle to the ModernUI_SmartPanel control
- [in] **PanelIndex** - integer value of the panel index to set as the current active panel
- [in] **bNotify** - post notification that current panel has changed

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUISmartPanelSetCurrentPanel, hSP, 0
```

See Also

MUISmartPanelGetCurrentPanel, MUISmartPanelCurrentPanelIndex

MUISmartPanelSetIsDlgMsgVar

MUISmartPanelSetIsDlgMsgVar, hWin:[MUIWND](#), lpVar:[LPMUIVALUE](#)

Specifies a variable that will be used during a message event loop for use with [IsDialogMessage](#). **lpVar** points to an address of a variable that will hold the handle to the current dialog panel.

Parameters

- [in] **hWin** - handle to the ModernUI_SmartPanel control
- [out] **lpVar** - pointer to variable to store the current panel handle (HWND)

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
.data
dwCurrentPanelHandle DD 0 ; global variable to store current panel handle
```

```
Invoke MUISmartPanelSetIsDlgMsgVar, hSP1, Addr dwCurrentPanelHandle
```

See Also

[MUISmartPanelRegisterPanel](#)

MUISmartPanelSetPanelParam

MUISmartPanelSetPanelParam

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref: , :ref: '
```

MUISmartPanelSetProperty

MUISmartPanelSetProperty, hWin:[MUIWND](#), Property:[MUIPROPERTY](#), PropertyValue:[MUIPROPERTYVALUE](#)

Sets the value of a property in a ModernUI_SmartPanel control. See [ModernUI_SmartPanel Properties](#) for details on the properties available

Parameters

- [in] **hWin** - handle to the ModernUI_SmartPanel control
- [in] **Property** - the property to set. See [ModernUI_SmartPanel Properties](#) for details on the properties available

- [in] **PropertyValue** - the value to set the property to

Return

Returns the previously set value for the property, or NULL otherwise

Example

```
Invoke MUISmartPanelSetProperty, hSP, @SmartPanelBackColor, MUI_RGBCOLOR(240,240,240)
```

See Also

MUISmartPanelGetProperty, ModernUI_SmartPanel Properties

Function	Description
<i>MUISmartPanelCreate</i>	Creates a new ModernUI_SmartPanel control
<i>MUISmartPanelCurrentPanelIndex</i>	Gets the current panel's index
<i>MUISmartPanelGetCurrentPanel</i>	Gets handle (HWND) to current active panel
<i>MUISmartPanelGetPanelParam</i>	Get IParam of panel - custom user data
<i>MUISmartPanelGetProperty</i>	Gets the value of a property
<i>MUISmartPanelNextPanel</i>	Moves to next panel that is registered and shows it
<i>MUISmartPanelNotifyCallback</i>	User specified callback for notifications
<i>MUISmartPanelPrevPanel</i>	Moves to previous panel that is registered and shows it
<i>MUISmartPanelRegister</i>	Registers a window class for the ModernUI_SmartPanel
<i>MUISmartPanelRegisterPanel</i>	Registers a dialog window (HWND) with the SmartPanel
<i>MUISmartPanelSetCurrentPanel</i>	Sets the current active panel
<i>MUISmartPanelSetIsDlgMsgVar</i>	Variable to receive current panel handle
<i>MUISmartPanelSetPanelParam</i>	Sets the IParam of a registered panel - custom user data
<i>MUISmartPanelSetProperty</i>	Sets the value of a property

3.7.2 ModernUI_SmartPanel Messages

MUISPM_GETCURRENTPANEL

MUISPM_GETCURRENTPANEL EQU WM_USER+1758

A custom windows message for the ModernUI_SmartPanel control that gets the current panel's index

Parameters

- **wParam** - NULL
- **lParam** - NULL

Return

Returns panel index

Example

```
Invoke SendMessage, hSmartPanel, MUISPM_GETCURRENTPANEL, NULL, NULL
```

See Also

MUISPM_SETCURRENTPANEL, MUISmartPanelSetCurrentPanel, MUISmartPanelGetCurrentPanel, MUISmartPanelCurrentPanelIndex

MUISPM_GETPANELPARAM

MUISPM_GETPANELPARAM EQU WM_USER+1753

A custom windows message for the ModernUI_SmartPanel control that gets the lParam of a panel which is custom user data

Parameters

- **wParam** - panel id to get panel's lParam for
- **lParam** - NULL

Return

Returns panel lParam value if successful, or NULL otherwise

Example

```
Invoke SendMessage, hSmartPanel, MUISPM_GETPANELPARAM, 1, NULL
```

See Also

[*MUISPM_SETPANELPARAM*](#)

MUISPM_GETTOTALPANELS

MUISPM_GETTOTALPANELS EQU WM_USER+1755

A custom windows message for the ModernUI_SmartPanel control that returns total panels registered

Parameters

- **wParam** - NULL
- **lParam** - NULL

Return

Returns total panels registered, or -1 if no panels registered

Example

```
Invoke SendMessage, hSmartPanel, MUISPM_GETTOTALPANELS, NULL, NULL
```

See Also

[*MUISPM_GETCURRENTPANEL*](#), [*MUISPM_SETCURRENTPANEL*](#), [*MUISmartPanelSetCurrentPanel*](#), [*MUISmartPanelGetCurrentPanel*](#), [*MUISmartPanelCurrentPanelIndex*](#)

MUISPM_NEXTPANEL

MUISPM_NEXTPANEL EQU WM_USER+1757

A custom windows message for the ModernUI_SmartPanel control that moves to next panel that is registered and shows it

Parameters

- **wParam** - NULL
- **lParam** - NULL

Return

None

Example

```
Invoke SendMessage, hSmartPanel, MUISPM_NEXTPANEL, NULL, NULL
```

See Also

MUISmartPanelNextPanel, MUISPM_PREVPANEL, MUISmartPanelPrevPanel

MUISPM_PREVPANEL

MUISPM_PREVPANEL EQU WM_USER+1756

A custom windows message for the ModernUI_SmartPanel control that moves to previous panel that is registered and shows it

Parameters

- **wParam** - NULL
- **lParam** - NULL

Return

None

Example

```
Invoke SendMessage, hSmartPanel, MUISPM_PREVPANEL, NULL, NULL
```

See Also

MUISmartPanelPrevPanel, MUISPM_NEXTPANEL, MUISmartPanelNextPanel

MUISPM_REGISTERPANEL

MUISPM_REGISTERPANEL EQU WM_USER+1760

A custom windows message for the ModernUI_SmartPanel control that registers a dialog window (HWND) with the ModernUI_SmartPanel control

Parameters

- **wParam** - Resource id of Dialog to register with the ModernUI_SmartPanel control
- **lParam** - Address of the Dialog's main procedure

Return

Returns handle of registered dialog HWND if successful, or NULL otherwise

Example

```
InfoDialogProc PROTO :HWND, :UINT, :WPARAM, :LPARAM
TestDialogProc PROTO :HWND, :UINT, :WPARAM, :LPARAM

.const
IDD_INFO_DIALOG EQU 2000 ; resource id assigned to dialog
IDD_TEST_DIALOG EQU 3000 ; resource id assigned to dialog
```

```
InfoDialogProc PROC hWin:HWND, uMsg:UINT, wParam:WPARAM, lParam:LPARAM
    mov eax, uMsg
    .IF eax == WM_INITDIALOG

    .ELSEIF eax == WM_COMMAND

    .ELSEIF eax==WM_CLOSE
        Invoke DestroyWindow, hWin

    .ELSE
        mov eax, FALSE
        ret
    .ENDIF
    mov eax, TRUE
    ret
InfoDialogProc ENDP

TestDialogProc PROC hWin:HWND, uMsg:UINT, wParam:WPARAM, lParam:LPARAM
    mov eax, uMsg
    .IF eax == WM_INITDIALOG

    .ELSEIF eax == WM_COMMAND

    .ELSEIF eax==WM_CLOSE
        Invoke DestroyWindow, hWin

    .ELSE
        mov eax, FALSE
        ret
    .ENDIF
    mov eax, TRUE
    ret
TestDialogProc ENDP
```

```
; Register dialog panels with our Modern_SmartPanel control
Invoke SendMessage, hSP, MUISPM_REGISTERPANEL, IDD_INFO_DIALOG, Addr InfoDialogProc
Invoke SendMessage, hSP, MUISPM_REGISTERPANEL, IDD_TEST_DIALOG, Addr TestDialogProc

Invoke SendMessage, hSP, MUISPM_SETCURRENTPANEL, 0, 0 ; set to first registered panel
```

See Also

MUISPM_SETCURRENTPANEL, MUISPM_GETCURRENTPANEL

MUISPM_SETCURRENTPANEL

MUISPM_SETCURRENTPANEL EQU WM_USER+1759

A custom windows message for the ModernUI_SmartPanel control that sets the current active panel

Parameters

- **wParam** - panel id to select
- **lParam** - send notify message: TRUE or FALSE

Return

Returns the index of the previously selected panel if successful or -1 otherwise.

Example

```
Invoke SendMessage, hSmartPanel, MUISPM_SETCURRENTPANEL, 1, TRUE
```

See Also

MUISPM_GETCURRENTPANEL, *MUISmartPanelGetCurrentPanel*, *MUISmartPanelSetCurrentPanel*, *MUISmartPanelCurrentPanelIndex*

MUISPM_SETISDLGMSGVAR

MUISPM_SETISDLGMSGVAR EQU WM_USER+1754

A custom windows message for the ModernUI_SmartPanel control that uses the variable pointed to by **wParam** to store the current panel handle of the ModernUI_SmartPanel control for use with [IsDialogMessage](#) function

Parameters

- **wParam** - Address of variable to receive the current panel handle (HWND)
- **lParam** - NULL

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
.data
dwCurrentPanelHandle DD 0 ; global variable to store current panel handle
```

```
Invoke SendMessage, hSmartPanel, MUISPM_SETISDLGMSGVAR, Addr dwCurrentPanelHandle,   
↳ NULL
```

See Also

MUISmartPanelRegisterPanel

MUISPM_SETPANELPARAM

MUISPM_SETPANELPARAM EQU WM_USER+1752

A custom windows message for the ModernUI_SmartPanel control that sets the panel's lParam of a registered panel which is custom user data

Parameters

- **wParam** - panel id to set panel's lParam for
- **lParam** - value to set panel's lParam to

Return

Returns lParam value if successful, or NULL otherwise

Example

```
Invoke SendMessage, hSmartPanel, MUISPM_SETPANELPARAM, 1, 1234
```

See Also

MUISPM_GETPANELPARAM

Message	Description
<i>MUISPM_GETCURRENTPANEL</i>	Gets the current panel's index
<i>MUISPM_GETPANELPARAM</i>	Get lParam of panel - custom user data
<i>MUISPM_GETTOTALPANELS</i>	Gets total panels registered with the SmartPanel control
<i>MUISPM_NEXTPANEL</i>	Moves to next panel that is registered and shows it
<i>MUISPM_PREVPANEL</i>	Moves to previous panel that is registered and shows it
<i>MUISPM_REGISTERPANEL</i>	Registers a dialog window (HWND) with the SmartPanel
<i>MUISPM_SETCURRENTPANEL</i>	Sets the current active panel
<i>MUISPM_SETISDLGMSGVAR</i>	Variable to receive current panel handle
<i>MUISPM_SETPANELPARAM</i>	Sets the lParam of a registered panel - custom user data

3.7.3 ModernUI_SmartPanel Properties

Property	Type
<i>@SmartPanelPanelsColor</i>	MUICOLORRGB
<i>@SmartPanelBorderColor</i>	MUICOLORRGB
<i>@SmartPanelNotifications</i>	BOOL
<i>@SmartPanelNotifyCallback</i>	POINTER
<i>@SmartPanelDllInstance</i>	MUIVALUE
<i>@SmartPanelParam</i>	MUIVALUE

ModernUI_SmartPanel Property Descriptions

@SmartPanelPanelsColor

Back color (*MUICOLORRGB*) for registered panels. If set to -1 then uses system default. Default is -1

@SmartPanelBorderColor

Border color (*MUICOLORRGB*) for registered panels. If set to -1 then no border. Default value is -1

@SmartPanelNotifications

Enable or disable notifications via *WM_NOTIFY* or *@SmartPanelNotifyCallback*. Default is TRUE

@SmartPanelNotifyCallback

Address of custom notification callback to use instead of *WM_NOTIFY*. If set to NULL uses *WM_NOTIFY*. Default is NULL

@SmartPanelDllInstance

Instance value for use in dll. Future use.

@SmartPanelParam

Custom user defined value to assign to ModernUI_SmartPanel control.

3.8 ModernUI_Spinner



The ModernUI_Spinner is a control typically used when loading, pre-loading or processing something and to hint or indicate to the user something is occurring.

3.8.1 ModernUI_Spinner Functions

MUISpinnerAddFrame

MUISpinnerAddFrame, hWin:*MUIWND*, ImageHandleType:*MUIIT*, hImage:*MUIIMAGE*

Adds an image frame to the spinner control

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **ImageHandleType** - type of image used in **hImage**: MUIIT_NONE, MUIIT_BMP, MUIIT_ICO, or MUIIT_PNG
- [in] **hImage** - handle to image

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUISpinnerAddFrame, hSpinner, MUI_PNG, pPngImage
```

See Also

MUISpinnerAddFrames, MUISpinnerLoadFrame, MUISpinnerLoadFrames, MUISpinnerLoadImage

MUISpinnerAddFrames

MUISpinnerAddFrames, hWin:*MUIWND*, Count:*MUIVALUE*, ImageHandleType:*MUIIT*, lpArrayImageHandles:*POINTER*

Process an array of image handles and add them to the spinner control as image frames

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **Count** - amount of handles in the array pointed to by **lpArrayImageHandles**
- [in] **ImageHandleType** - type of images used in array of handles **lpArrayImageHandles**: MUIIT_NONE, MUIIT_BMP, MUIIT_ICO, or MUIIT_PNG
- [in] **lpArrayImageHandles** - pointer to array of image handles

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUISpinnerAddFrames, hSpinner, 10, MUI_PNG, Addr aPngHandles
```

See Also

MUISpinnerAddFrame, MUISpinnerLoadFrame, MUISpinnerLoadFrames, MUISpinnerLoadImage

MUISpinnerAddImage

MUISpinnerAddImage, hWin:*MUIWND*, hImage:*GPIMAGE*, NoFramesToCreate::*MUIVALUE*, bReverse:BOOL

Add a single image (previously loaded), to rotate and create the animation frames for the spinner from. Note: **hImage** must be a **GDI+** image.

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **hImage** - handle to the image to add
- [in] **NoFramesToCreate** - no of frames to create from single image
- [in] **bReverse** - if TRUE reverses order of created frames

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUISpinnerAddImage, pImageSpinCycle, 22, FALSE
```

See Also

MUISpinnerLoadImage, MUISpinnerLoadFrame, MUISpinnerLoadFrames, MUISpinnerAddFrame, MUISpinnerAddFrames

MUISpinnerAddSpriteSheet

MUISpinnerAddSpriteSheet, hWin:*MUIWND*, SpriteCount:*MUIVALUE*, ImageHandleType:*MUIIT*, hImage-SpriteSheet:*MUIIMAGE*, bReverse:BOOL

Adds a long (wide) image strip from a previously loaded image, to use as the frames of the spinner animation. The width of each frame is calculated by dividing the length of the spritesheet image by **SpriteCount**

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **SpriteCount** - no of frames to split the spritesheet into
- [in] **ImageHandleType** - type of image used in **hImageSpriteSheet**: MUIIT_NONE, MUIIT_BMP, MUIIT_ICO, or MUIIT_PNG
- [in] **hImageSpriteSheet**
- [in] **bReverse** - if TRUE reverses order of created frames

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUISpinnerAddSpriteSheet, hSpinner, 22, MUI_PNG, pImageSpriteSheet, FALSE
```

See Also

MUISpinnerLoadSpriteSheet

MUISpinnerCreate

MUISpinnerCreate, hWndParent:*MUIWND*, X:*MUIVALUE*, Y:*MUIVALUE*, nWidth:*MUIVALUE*, nHeight:*MUIVALUE*, ResourceID:*RESID*, Style:*MUIVALUE*

Creates a new ModernUI_Spinner control

Parameters

- [in] **hWndParent** - parent window of control
- [in] **X** - x coordinate of the control
- [in] **Y** - y coordinate of the control
- [in] **nWidth** - width of the control
- [in] **nHeight** - height of the control
- [in] **ResourceID** - resource id of the control
- [in] **Style** - can be combination of style flags, see *ModernUI_Spinner Style Flags* for details

Return

Returns handle to newly created ModernUI_Spinner control (MUIWND) if successful, or NULL otherwise

ModernUI_Spinner Style Flags

- MUISPNS_HAND - Show a hand instead of an arrow when mouse moves over spinner

Example

```
Invoke MUISpinnerCreate, hWin, 100, 100, 64, 64, IDC_SPINNER, 0
```

See Also

MUISpinnerRegister, *MUISpinnerGetProperty*, *MUISpinnerSetProperty*

MUISpinnerDisable

MUISpinnerDisable, hWin:*MUIWND*

Disables the spinner animation and hides the ModernUI_Spinner control

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control

Return

None

Example

```
Invoke MUISpinnerDisable, hSpinner
```

See Also

MUISpinnerEnable, MUISpinnerPause, MUISpinnerResume

MUISpinnerEnable

MUISpinnerEnable, hWin:*MUIWND*

Enables the spinner animation and shows the ModernUI_Spinner control

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control

Return

None

Example

```
Invoke MUISpinnerEnable, hSpinner
```

See Also

MUISpinnerDisable, MUISpinnerPause, MUISpinnerResume

MUISpinnerGetProperty

MUISpinnerGetProperty, hWin:*MUIWND*, Property:*MUIPROPERTY*

Get the value of a property in a ModernUI_SmartPanel control. See *ModernUI_Spinner Properties* for details on the properties available

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **Property** - the property to get. See *ModernUI_Spinner Properties* for details on the properties available

Return

Returns the value of the property or NULL otherwise

Example

```
LOCAL BackColor:DWORD

Invoke MUISpinnerGetProperty, hSpinner, @SpinnerBackColor
mov BackColor, eax
```

See Also

MUISpinnerSetProperty, ModernUI_Spinner Properties

MUISpinnerLoadFrame

MUISpinnerLoadFrame, hWin:[MUIWND](#), ImageHandleType:[MUIIT](#), idResImage:[RESID](#)

Loads a resource as an image frame to the spinner control

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **ImageHandleType** - type of image to load in **idResImage**: MUIIT_NONE, MUIIT_BMP, MUIIT_ICO, or MUIIT_PNG
- [in] **idResImage** - resource id of image to load

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUISpinnerLoadFrame, hSpinner, MUI_BMP, BMP_TESTER
```

See Also

[MUISpinnerAddFrames](#), [MUISpinnerAddFrames](#), [MUISpinnerLoadFrames](#), [MUISpinnerLoadImage](#)

MUISpinnerLoadFrames

MUISpinnerLoadFrames, hWin:[MUIWND](#), Count:[MUIVALUE](#), ImageHandleType:[MUIIT](#), lpArrayResourceIDs:[POINTER](#)

Process an array of resource ids and load the resource and add them to the spinner control as image frames

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **Count** - amount of handles in the array pointed to by **lpArrayResourceIDs**
- [in] **ImageHandleType** - type of images used in array of resource ids **lpArrayResourceIDs**: MUIIT_NONE, MUIIT_BMP, MUIIT_ICO, or MUIIT_PNG
- [in] **lpArrayResourceIDs** - pointer to array of image resource ids

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUISpinnerLoadFrames, hSpinner, 12, MUI_BMP, Addr aBitmapResources
```

See Also

[MUISpinnerAddFrame](#), [MUISpinnerAddFrames](#), [MUISpinnerLoadFrame](#), [MUISpinnerLoadImage](#)

MUISpinnerLoadImage

MUISpinnerLoadImage, hWin:[MUIWND](#), idResImage:[RESID](#), NoFramesToCreate::[MUIVALUE](#), bReverse:BOOL

Loads a single image from a resource, to rotate and create the animation frames for the spinner from. Note: **idResImage** must be a **PNG** stored in RC_DATA format

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **idResImage** - resource id of image to load
- [in] **NoFramesToCreate** - no of frames to create from single image
- [in] **bReverse** - if TRUE reverses order of created frames

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUISpinnerLoadImage, hSpinner, PNG_SPINCYLE, 22, FALSE
```

See Also

MUISpinnerAddImage, MUISpinnerLoadFrame, MUISpinnerLoadFrames, MUISpinnerAddFrame, MUISpinnerAddFrames

MUISpinnerLoadSpriteSheet

MUISpinnerLoadSpriteSheet, hWin:*MUIWND*, SpriteCount:*MUIVALUE*, ImageHandleType:*MUIIT*, idResSpriteSheet:*RESID*, bReverse:BOOL

Loads a long (wide) image strip from a resource id, to use as the frames of the spinner animation. The width of each frame is calculated by dividing the length of the spritesheet image by **SpriteCount**

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **SpriteCount** - no of frames to split the spritesheet into
- [in] **ImageHandleType** - type of image used in resource **idResSpriteSheet**: MUIIT_NONE, MUIIT_BMP, MUIIT_ICO, or MUIIT_PNG
- [in] **idResSpriteSheet**
- [in] **bReverse** - if TRUE reverses order of created frames

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUISpinnerLoadSpriteSheet, hSpinner, 22, MUI_BMP, BMP_SPINSPRITESHEET, FALSE
```

See Also

MUISpinnerAddSpriteSheet

MUISpinnerPause

MUISpinnerPause, hWin:*MUIWND*

Pause the spinner animation that has been started via *MUISpinnerEnable*

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control

Return

None

Example

```
Invoke MUISpinnerPause, hSpinner
```

See Also

MUISpinnerResume, MUISpinnerEnable, MUISpinnerDisable

MUISpinnerRegister**MUISpinnerRegister**

Registers the ModernUI_Spinner class, which allows the control to be created via dialog resources or via CreateWindowEx and specifying the classname 'ModernUI_Spinner'. If using the RadASM custom class control instead of a the ModernUI RadASM Design Time Controls specify 'ModernUI_Spinner' as the classname.

Parameters

None

Return

None

Example

```
Invoke MUISpinnerRegister
```

See Also

MUISpinnerCreate, MUISpinnerGetProperty, MUISpinnerSetProperty

MUISpinnerReset

MUISpinnerReset, hWin:*MUIWND*

Reset the spinner's current frame of the ModernUI_Spinner control animation

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control

Return

None

Example

```
Invoke MUISpinnerReset, hSpinner
```

See Also

MUISpinnerEnable, MUISpinnerDisable

MUISpinnerResume

MUISpinnerResume, hWin:*MUIWND*

Resume the spinner after it has been paused by *MUISpinnerPause*

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke MUISpinnerResume, hSpinner
```

See Also

MUISpinnerPause, *MUISpinnerEnable*, *MUISpinnerDisable*

MUISpinnerSetProperty

MUISpinnerSetProperty, hWin:*MUIWND*, Property:*MUIPROPERTY*, PropertyValue:*MUIPROPERTYVALUE*

Sets the value of a property in a ModernUI_Spinner control. See *ModernUI_Spinner Properties* for details on the properties available

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **Property** - the property to set. See *ModernUI_Spinner Properties* for details on the properties available
- [in] **PropertyValue** - the value to set the property to

Return

Returns the previously set value for the property, or NULL otherwise

Example

```
Invoke MUISpinnerSetProperty, hSpinner, @SpinnerBackColor, MUI_RGBCOLOR(240,240,240)
```

See Also

MUISpinnerGetProperty, *ModernUI_Spinner Properties*

MUISpinnerSpeed

MUISpinnerSpeed, hWin:*MUIWND*, MillisecSpeed:MUIVALUE

Sets the spinner animation speed (in milliseconds)

Parameters

- [in] **hWin** - handle to the ModernUI_Spinner control
- [in] **MillisecSpeed**

Return

None

Example

```
Invoke MUISpinnerSpeed, hSpinner, 100
```

See Also

MUISpinnerGetProperty, MUISpinnerSetProperty

Function	Description
<i>MUISpinnerAddFrame</i>	Adds an image handle (previously loaded) to the control
<i>MUISpinnerAddFrames</i>	Adds an array of image handles (previously loaded)
<i>MUISpinnerAddImage</i>	Adds a single image handle (previously loaded), to rotate
<i>MUISpinnerAddSpriteSheet</i>	Adds a spritesheet - an imagestrip of animation frames
<i>MUISpinnerCreate</i>	Creates a new ModernUI_Spinner control
<i>MUISpinnerDisable</i>	Disable and hide the spinner animation
<i>MUISpinnerEnable</i>	Enable and show the spinner animation
<i>MUISpinnerGetProperty</i>	Gets the value of a property
<i>MUISpinnerLoadFrame</i>	Loads an image from a resource and adds to the control
<i>MUISpinnerLoadFrames</i>	Loads an array of resources and adds the frames
<i>MUISpinnerLoadImage</i>	Loads a single image from a resource, to rotate
<i>MUISpinnerLoadSpriteSheet</i>	Loads a spritesheet from a resource
<i>MUISpinnerPause</i>	Pause the spinner animation
<i>MUISpinnerRegister</i>	Registers a window class for the ModernUI_Spinner
<i>MUISpinnerReset</i>	Reset the spinner's current frame
<i>MUISpinnerResume</i>	Resume the spinner animation
<i>MUISpinnerSetProperty</i>	Sets the value for a property
<i>MUISpinnerSpeed</i>	Set the spinner animation speed (in milliseconds)

3.8.2 ModernUI_Spinner Messages

MUISPNM_ADDFRAME

MUISPNM_ADDFRAME EQU WM_USER+1752

A custom windows message for the ModernUI_Spinner that adds an image handle (previously loaded) to the control

Parameters

- **wParam** - the spinner image type: MUISPIT_NONE, MUISPIT_BMP, MUISPIT_ICO or MUISPIT_PNG
- **lParam** - handle to the image to add to the spinner

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke SendMessage, hSpinner, MUISPNM_ADDFRAME, MUISPIT_BMP, hBitmapImage
```

See Also

MUISpinnerAddFrame, MUISPNM_LOADFRAME, MUISpinnerLoadFrame

MUISPNM_DISABLE

MUISPNM_DISABLE EQU WM_USER+1749

A custom windows message for the ModernUI_Spinner that disables and hides the spinner animation

Parameters

- **wParam** - NULL
- **lParam** - NULL

Return

None

Example

```
Invoke SendMessage, hSpinner, MUISPNM_DISABLE, NULL, NULL
```

See Also

[*MUISPNM_ENABLE*](#), [*MUISPNM_RESUME*](#)

MUISPNM_ENABLE

MUISPNM_ENABLE EQU WM_USER+1750

A custom windows message for the ModernUI_Spinner that enables and shows the spinner animation

Parameters

- **wParam** - NULL
- **lParam** - NULL

Return

None

Example

```
Invoke SendMessage, hSpinner, MUISPNM_ENABLE, NULL, NULL
```

See Also

[*MUISPNM_DISABLE*](#), [*MUISPNM_PAUSE*](#)

MUISPNM_LOADFRAME

MUISPNM_LOADFRAME EQU WM_USER+1751

A custom windows message for the ModernUI_Spinner that loads an image from a resource and adds to the ModernUI_Spinner control

Parameters

- **wParam** - the spinner image type: MUISPIT_NONE, MUISPIT_BMP, MUISPIT_ICO or MUISPIT_PNG
- **lParam** - resource id of the image to load and add to the spinner

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
Invoke SendMessage, hSpinner, MUISPNM_LOADFRAME, MUISPIT_PNG, pPngImage
```

See Also

MUISpinnerLoadFrame, MUISPNM_ADDFRAME, MUISpinnerAddFrame

MUISPNM_PAUSE

MUISPNM_PAUSE EQU WM_USER+1747

A custom windows message for the ModernUI_Spinner that pauses the spinner animation

Parameters

- **wParam** - NULL
- **lParam** - NULL

Return

None

Example

```
Invoke SendMessage, hSpinner, MUISPNM_PAUSE, NULL, NULL
```

See Also

MUISPNM_RESUME, MUISPNM_ENABLE, MUISPNM_DISABLE

MUISPNM_RESET

MUISPNM_RESET EQU WM_USER+1748

A custom windows message for the ModernUI_Spinner that reset the spinner's current frame

Parameters

- **wParam** - NULL
- **lParam** - NULL

Return

None

Example

```
Invoke SendMessage, hSpinner, MUISPNM_RESET, NULL, NULL
```

See Also

MUISPNM_ENABLE, MUISPNM_DISABLE

MUISPNM_RESUME

MUISPNM_RESUME EQU WM_USER+1746

A custom windows message for the ModernUI_Spinner that resumes the spinner animation

Parameters

- **wParam** - NULL
- **lParam** - NULL

Return

None

Example

```
Invoke SendMessage, hSpinner, MUISPNM_RESUME, NULL, NULL
```

See Also

MUISPNM_PAUSE, MUISPNM_ENABLE, MUISPNM_DISABLE

MUISPNM_SPEED

MUISPNM_SPEED EQU WM_USER+1745

A custom windows message for the ModernUI_Spinner that sets the spinner animation speed (in milliseconds)

Parameters

- **wParam** - speed in milliseconds for spinner animation
- **lParam** - NULL

Return

None

Example

```
Invoke SendMessage, hSpinner, MUISPNM_SPEED, 100, NULL
```

See Also

MUISPNM_PAUSE, MUISPNM_RESUME, MUISPNM_ENABLE, MUISPNM_DISABLE

Message	Description
<i>MUISPNM_ADDFRAME</i>	Adds an image handle (previously loaded) to the control
<i>MUISPNM_DISABLE</i>	Disable and hide the spinner animation
<i>MUISPNM_ENABLE</i>	Enable and show the spinner animation
<i>MUISPNM_LOADFRAME</i>	Loads an image from a resource and adds to the control
<i>MUISPNM_PAUSE</i>	Pause the spinner animation
<i>MUISPNM_RESET</i>	Reset the spinner's current frame
<i>MUISPNM_RESUME</i>	Resume the spinner animation
<i>MUISPNM_SPEED</i>	Set the spinner animation speed (in milliseconds)

3.8.3 ModernUI_Spinner Properties

Property	Type
@SpinnerBackColor	MUICOLORRGB
@SpinnerSpeed	MUIVALUE
@SpinnerDllInstance	HINSTANCE

ModernUI_Spinner Property Descriptions

@SpinnerBackColor

Background color (*MUICOLORRGB*) of the ModernUI_Spinner control. By default will try to obtain the color of the parent's background color otherwise will default to GetSysColor value returned by `COLOR_WINDOW`

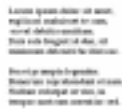
@SpinnerSpeed

A value (*MUIVALUE*) indicating speed in milliseconds of the ModernUI_Spinner control frame animation. Default speed is 80 milliseconds

@SpinnerDllInstance

Used for loading resources when ModernUI_Spinner control is used in a dll. Default is NULL

3.9 ModernUI_Text



The ModernUI_Text is a static display of text. However it provides ease of use and customizable features like a small set of font families (Arial, Courier New, Segoe UI, Tahoma, Times New Roman and Verdana), font size (7pt-32pt), font effects (like bold, italic, underline), text color, back color and more.

These features allow you set the desired font style and size just with a few style flags when creating the ModernUI_Text control. Colors and other properties can be set at run time, without having to write extra code to handle `WM_CTLCOLORSTATIC` and/or creating fonts to assign to the control.

3.9.1 ModernUI_Text Functions

MUITextCreate

MUITextCreate, hWndParent:*MUIWND*, lpzText:LPSTR, X:*MUIVALUE*, Y:*MUIVALUE*, nWidth:*MUIVALUE*, nHeight:*MUIVALUE*, ResourceID:*RESID*, Style:*MUIVALUE*

Creates a new ModernUI_Text control.

Parameters

- [in] **hWndParent** - parent window of control
- [in] **lpzText** - text to display
- [in] **X** - x coord of control

- [in] **Y** - y coord of control
- [in] **nWidth** - width of control
- [in] **nHeight** - height of control
- [in] **ResourceID** - resource id of control
- [in] **Style** - can be combination of style flags, see *ModernUI_Text Style Flags* for details

Return

Returns handle to newly created ModernUI_Text control (MUIWND) if successful, or NULL otherwise

ModernUI_Text Style Flags

The following combination of flags can be specified for **Style**:

Font size style flags (a single flag from this group):

- MUIITS_7PT - 7pt
- MUIITS_8PT - 8pt
- MUIITS_9PT - 9pt
- MUIITS_10PT - 10pt
- MUIITS_11PT - 11pt
- MUIITS_12PT - 12pt
- MUIITS_13PT - 13pt
- MUIITS_14PT - 14pt
- MUIITS_15PT - 15pt
- MUIITS_16PT - 16pt
- MUIITS_18PT - 18pt
- MUIITS_20PT - 20pt
- MUIITS_22PT - 22pt
- MUIITS_24PT - 24pt
- MUIITS_28PT - 28pt
- MUIITS_32PT - 32pt

Font family style flags (a single flag from this group):

- MUIITS_FONT_DIALOG - Use font that dialog is using
- MUIITS_FONT_SEGOE - Segoe UI font
- MUIITS_FONT_TAHOMA - Tahoma font
- MUIITS_FONT_ARIAL - Arial font
- MUIITS_FONT_TIMES - Times New Roman font
- MUIITS_FONT_COURIER - Courier New font
- MUIITS_FONT_VERDANA - Verdana font

Text alignment style flags (a single flag from this group):

- MUIITS_ALIGN_LEFT - left align text

- MUIITS_ALIGN_RIGHT - right align text
- MUIITS_ALIGN_CENTER - center text
- MUIITS_ALIGN_JUSTIFY - justify text

Font special style flags (combination of flags allowed):

- MUIITS_FONT_NORMAL - No bold, italic or underline
- MUIITS_FONT_BOLD - Bold text
- MUIITS_FONT_ITALIC - Italic text
- MUIITS_FONT_UNDERLINE - Underline text

Misc options style flags (combination of flags allowed):

- MUIITS_SINGLELINE - Single line of text, otherwise is multi line
- MUIITS_HAND - Show a hand instead of an arrow when mouse moves over text
- MUIITS_LORUMIPSUM - Show lorem ipsum in text box - for demo purposes etc
- MUIITS_UTF8 - Text is utf8 format
- MUIITS_HTMLCODE - Text has htmlcode tags to decode
- MUIITS_BBCODE - Text has bbcode tags to decode

Example

```
LOCAL TextStyle:DWORD

mov TextStyle, MUIITS_22PT or MUIITS_FONT_SEGOE or MUIITS_ALIGN_LEFT or MUIITS_FONT_BOLD
Invoke MUITextCreate, hWin, Addr szText, 10, 10, 400, 400, IDC_TXT1, TextStyle
```

See Also

[MUITextRegister](#), [MUITextSetProperty](#), [MUITextGetProperty](#)

MUITextGetProperty

MUITextGetProperty, hWin:[MUIWND](#), Property:[MUIPROPERTY](#)

Get the value of a property in the ModernUI_Text control. See [ModernUI_Text Properties](#) for details on the properties available

Parameters

- [in] **hWin** - handle to the ModernUI_Text control
- [in] **Property** - the property to get. See [ModernUI_Text Properties](#) for details on the properties available

Return

Returns the value of the property or NULL otherwise

Example

```
LOCAL dwTextColor:DWORD

Invoke MUITextGetProperty, hTxt1, @TextColor
mov dwTextColor, eax
```

See Also

MUITextSetProperty, *ModernUI_Text Properties*

MUITextRegister

MUITextRegister

Registers the ModernUI_Text class, which allows the control to be created via dialog resources or via [CreateWindowEx](#) and specifying the classname 'ModernUI_Text'. If using the RadASM custom class control instead of a the ModernUI RadASM Design Time Controls specify 'ModernUI_Text' as the classname

Parameters

None

Return

None

Example

```
Invoke MUITextRegister
```

See Also

MUITextCreate, *MUITextGetProperty*, *MUITextSetProperty*

MUITextSetBufferSize

MUITextSetBufferSize

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref: ' , :ref: '

MUITextSetProperty

MUITextSetProperty, hWin:*MUIWND*, Property:*MUIPROPERTY*, PropertyValue:*MUIPROPERTYVALUE*

Sets the value of a property in the ModernUI_Text control. See *ModernUI_Text Properties* for details on the properties available

Parameters

- [in] **hWin** - handle to the ModernUI_Text control
- [in] **Property** - the property to set. See *ModernUI_Text Properties* for details on the properties available

- [in] **PropertyValue** - the value to set the property to

Return

Returns the previously set value for the property, or NULL otherwise

Example

```
Invoke MUITextSetProperty, hTxt1, @TextColor, MUI_RGBCOLOR(48,48,48)
```

See Also

MUITextGetProperty, *ModernUI_Text Properties*

Function	Description
<i>MUITextCreate</i>	Creates a new ModernUI_Text control
<i>MUITextGetProperty</i>	Gets the value of a property
<i>MUITextRegister</i>	Registers a window class for the ModernUI_Text
<i>MUITextSetBufferSize</i>	Sets internal buffer size for text strings
<i>MUITextSetProperty</i>	Sets the value of a property

3.9.2 ModernUI_Text Properties

Property	Type
<i>@TextFont</i>	HFONT
<i>@TextColor</i>	MUI_COLORRGB
<i>@TextColorAlt</i>	MUI_COLORRGB
<i>@TextColorDisabled</i>	MUI_COLORRGB
<i>@TextBackColor</i>	MUI_COLORRGB
<i>@TextBackColorAlt</i>	MUI_COLORRGB
<i>@TextBackColorDisabled</i>	MUI_COLORRGB

ModernUI_Text Property Descriptions

@TextFont

Font used for ModernUI_Text control text

@TextColor

Text color (*MUI_COLORRGB*) of the ModernUI_Text control

@TextColorAlt

Text color (*MUI_COLORRGB*) when mouse moves over ModernUI_Text control

@TextColorDisabled

Text color (*MUI_COLORRGB*) when ModernUI_Text control is disabled

@TextBackColor

Background (*MUI_COLORRGB*) color of the ModernUI_Text control. For transparency use -1

@TextBackColorAlt

Background (*MUI_COLORRGB*) color when mouse moves over ModernUI_Text control

@TextBackColorDisabled

Background (*MUICOLORRGB*) color when ModernUI_Text control is disabled

3.10 ModernUI_Tooltip



The ModernUI_Tooltip is a small popup window that displays text. It can display a single line of text, similar to a standard win32 tooltip control, but can also display a title and additional lines of text below the title. The title text font can be different from the body text font. The back color and text color can be specified via the ModernUI_Tooltip's properties.

The tooltip position can also be customized to appear at the mouse cursor position, or above, below, to the right or to the left of the associated control, and can be offset by an x or y amount as well.

3.10.1 ModernUI_Tooltip Functions

MUITooltipCreate

MUITooltipCreate

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:`, :ref: `

MUITooltipGetProperty

MUITooltipGetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref: ‘
```

MUITooltipRegister

MUITooltipRegister

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref: ‘
```

MUITooltipSetProperty

MUITooltipSetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref: ‘
```

3.10.2 ModernUI_Tooltip Messages

3.10.3 ModernUI_Tooltip Properties

3.11 ModernUI_TrayMenu



The ModernUI_TrayMenu is a control to help create and manage system tray icons and menus that display when a user clicks on the system tray icon.

3.11.1 ModernUI_TrayMenu Functions

MUITrayIconCreate

MUITrayIconCreate

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayIconDestroy

MUITrayIconDestroy

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayIconSetTooltipText

MUITrayIconSetTooltipText

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayIconSetTrayIcon

MUITrayIconSetTrayIcon

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayIconSetTrayIconText

MUITrayIconSetTrayIconText

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayIconShowNotification

MUITrayIconShowNotification

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayMenuAssignMenu

MUITrayMenuAssignMenu

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayMenuChangeMenuItemState

MUITrayMenuChangeMenuItemState

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref:‘
```

MUITrayMenuCreate

MUITrayMenuCreate

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref:‘
```

MUITrayMenuDisableMenuItem

MUITrayMenuDisableMenuItem

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref:‘
```

MUITrayMenuEnableMenuItem

MUITrayMenuEnableMenuItem

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayMenuGetProperty

MUITrayMenuGetProperty

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayMenuHideTrayIcon

MUITrayMenuHideTrayIcon

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayMenuRegister

MUITrayMenuRegister

Parameters

- [in] 1

- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref: ‘
```

MUITrayMenuSetMenuItemText**MUITrayMenuSetMenuItemText****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref: ‘
```

MUITrayMenuSetProperty**MUITrayMenuSetProperty****Parameters**

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

```
:ref:‘ , :ref: ‘
```

MUITrayMenuSetTooltipText

MUITrayMenuSetTooltipText

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayMenuSetTrayIcon

MUITrayMenuSetTrayIcon

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayMenuSetTrayIconText

MUITrayMenuSetTrayIconText

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayMenuShowNotification

MUITrayMenuShowNotification

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

MUITrayMenuShowTrayIcon

MUITrayMenuShowTrayIcon

Parameters

- [in] 1
- [in] 2
- [in] 3

Return

Returns TRUE if successful, or FALSE otherwise

Example

```
**See Also**
```

:ref:‘, :ref:‘

3.11.2 ModernUI_TrayMenu Messages

3.11.3 ModernUI_TrayMenu Properties

The following topics cover more detail about some technical aspects of building the ModernUI library and also integrating the RadASM autocomplete / intellisense files, installing the design time ModernUI controls for RadASM and adding the UASM assembler to RadASM.

Contents:

Contents:

4.1 ModernUI x86 Build

The ModernUI Library and ModernUI Controls come with a RadASM project to help build the sources. However if you wish to build them manually here are the command line options you should use.

4.1.1 Building ModernUI Library

The ModernUI Library consists of two files:

- ModernUI.inc
- ModernUI.asm

Building with Microsoft MASM (ML.EXE):

```
ML.EXE /c /coff /Cp /nologo /I"X:\MASM32\Include" ModernUI.asm
```

X: is the drive letter where your MASM32 SDK includes files are located, replace with the appropriate drive letter for your installation, or modify the path for your installed location.

Linking with Microsoft Library Manager (LIB.EXE):

```
LIB *.obj /out:ModernUI.lib
```

4.1.2 Building ModernUI Controls

Each ModernUI Control consists of two files:

- ModernUI_Control.inc
- ModernUI_Control.asm

Building with Microsoft MASM (ML.EXE):

```
ML.EXE /c /coff /Cp /nologo /I"X:\MASM32\Include" ModernUI_Control.asm
```

**X: is the drive letter where your MASM32 SDK includes files are located, replace with the appropriate drive letter for your installation, or modify the path for your installed location.*

Linking with Microsoft Library Manager (LIB.EXE):

```
LIB *.obj /out:ModernUI_Control.lib
```

4.1.3 Debug x86 Builds

To build the ModernUI Library and/or a ModernUI Control with debug information, supply the additional flag options `/Zi /Zd` on the command line for MASM (ML.EXE) like so: .. code-block:: text

```
ML.EXE /c /coff /Cp /Zi /Zd /nologo /I"X:\MASM32\Include" ModernUI.asm
```

```
ML.EXE /c /coff /Cp /Zi /Zd /nologo /I"X:\MASM32\Include" ModernUI_Control.asm
```

**X: is the drive letter where your MASM32 SDK includes files are located, replace with the appropriate drive letter for your installation, or modify the path for your installed location.*

4.2 ModernUI x64 Build

The ModernUI Library and ModernUI Controls come with a RadASM project to help build the sources. However if you wish to build them manually here are the command line options you should use.

4.2.1 Building ModernUI Library

The ModernUI Library consists of two files:

- ModernUI.inc
- ModernUI.asm

Building with Microsoft UASM (UASM64.EXE):

```
UASM64.EXE /c -win64 -Zp8 /win64 /D_WIN64 /Cp /nologo /I"X:\UASM\Include" ModernUI.asm
```

**X: is the drive letter where your UASM SDK includes files are located, replace with the appropriate drive letter for your installation, or modify the path for your installed location.*

Linking with Microsoft Library Manager (LIB.EXE):

```
LIB /LIBPATH:"X:\UASM\lib\x64" *.obj /out:ModernUI.lib
```


X: is the drive letter where your UASM SDK 64bit library files are located, replace with the appropriate drive letter for your installation, or modify the path for your installed location.

4.2.2 Building ModernUI Controls

Each ModernUI Control consists of two files:

- ModernUI_Control.inc
- ModernUI_Control.asm

Building with Microsoft UASM (UASM64.EXE):

```
UASM64.EXE /c -win64 -Zp8 /win64 /D_WIN64 /Cp /nologo /I"X:\MASM32\Include" ModernUI_
↪Control.asm
```

**X: is the drive letter where your UASM64 SDK includes files are located, replace with the appropriate drive letter for your installation, or modify the path for your installed location.*

Linking with Microsoft Library Manager (LIB.EXE):

```
LIB /LIBPATH:"X:\UASM\lib\x64" *.obj /out:ModernUI_Control.lib
```

**X: is the drive letter where your UASM SDK 64bit library files are located, replace with the appropriate drive letter for your installation, or modify the path for your installed location.*

4.2.3 Debug x64 Builds

To build the ModernUI Library and/or a ModernUI Control with debug information, supply the additional flag options /Zi /Zd on the command line for UASM64 (UASM64.EXE) like so:

```
UASM64.EXE /c -win64 -Zp8 /win64 /D_WIN64 /Cp /nologo /Zi /Zd /I"X:\UASM\Include" ↪
↪ModernUI.asm
```

```
UASM64.EXE /c -win64 -Zp8 /win64 /D_WIN64 /Cp /nologo /Zi /Zd /I"X:\UASM\Include" ↪
↪ModernUI_Control.asm
```

**X: is the drive letter where your UASM SDK includes files are located, replace with the appropriate drive letter for your installation, or modify the path for your installed location.*

Linking with Microsoft Library Manager (LIB.EXE):

```
LIB /LIBPATH:"X:\UASM\lib\x64" *.obj /out:ModernUI_Control.lib
```

**X: is the drive letter where your UASM SDK 64bit library files are located, replace with the appropriate drive letter for your installation, or modify the path for your installed location.*

4.3 RadASM Support

RadASM auto-complete / intellisense api files and some design time ModernUI UI controls for RadASM are available to help with using the various ModernUI functions and controls.

4.3.1 RadASM auto-complete / intellisense

RadASM can support multiple languages. For developing for x86 it is recommended to use the [MASM32 assembler](#). For x64 development it is recommended to use the [UASM assembler](#).

MASM32 x86 assembler auto-complete / intellisense files:

- masmApiCall.api.txt
- masmApiConst.api.txt
- masmApiStruct.api.txt
- masmMessage.api.txt
- masmType.api.txt

UASM x64 assembler auto-complete / intellisense files:

- Uasm64ApiCall.api.txt
- Uasm64ApiConst.api.txt
- Uasm64ApiStruct.api.txt
- Uasm64Message.api.txt
- Uasm64Type.api.txt

Each `.api.txt` file can be opened and the contents copied and pasted into a corresponding `.api` file found under the RadASM folder for the particular assembler. Once RadASM is restarted it will read these files and provide the auto-complete / intellisense for ModernUI functions.

4.3.2 RadASM Design Time ModernUI Controls

A RAMUIControls.zip file is provided in the ModernUI repository in the release section that contains a collection of the design time ModernUI controls for RadASM. Each control is provided for in a separate dll file.

The current ModernUI controls that have a RadASM IDE control are:

- ModernUI_Button (RAMUIButton.dll)
- ModernUI_CaptionBar (RAMUICaptionBar.dll)
- ModernUI_Checkbox (RAMUICheckbox.dll)
- ModernUI_ProgressBar (RAMUIProgressBar.dll)
- ModernUI_ProgressDots (RAMUIProgressDots.dll)
- ModernUI_SmartPanel (RAMUISmartPanel.dll)
- ModernUI_Spinner (RAMUISpinner.dll)
- ModernUI_Text (RAMUIText.dll)

Installation

- Ensure RadASM is closed before installing the design time ModernUI controls.
- Copy the RAMUIxxxxxxx.dll files to your RadASM folder
- Edit **RadASM.ini** and add the controls to the section entitled `[CustCtrl]`. Each entry should be a unique number in sequential order.

Once completed it should look something like this:

```
[CustCtrl]
1=RAMUIButton.dll
2=RAMUICaptionBar.dll
3=RAMUICheckbox.dll
4=RAMUIProgressBar.dll
5=RAMUIProgressDots.dll
6=RAMUISmartPanel.dll
7=RAMUIText.dll
8=RAMUISpinner.dll
```

Usage

Once the design time ModernUI controls are installed (and RadASM is restarted) they will show up as icons in the dialog tools/toolbox toolbar. These provide a visual way of adding the ModernUI controls to your project and setting a few properties

It is important to note that the dll files that contain each ModernUI control is not a full version of the tool, merely enough to provide a visual representation of the tool when added to a dialog. You still have to include the appropriate library and include files AND call to register the ModernUI control before the dialog is created. Typically this registration should be done at the start of your program

For example if you wish to use the ModernUI_CaptionBar control and add it to a dialog, you should also place a call to register this control at start, something like:

```
start:

    Invoke GetModuleHandle, NULL
    mov hInstance, eax
    Invoke GetCommandLine
    mov CommandLine, eax
    Invoke InitCommonControls
    mov icc.dwSize, sizeof INITCOMMONCONTROLSEX
    mov icc.dwICC, ICC_COOL_CLASSES or ICC_STANDARD_CLASSES or ICC_WIN95_CLASSES
    Invoke InitCommonControlsEx, offset icc

    ; Register the control so that when the dialog that contains it is created
    ; it knows where to find the CaptionBar control.
    Invoke MUICaptionBarRegister

    Invoke WinMain, hInstance, NULL, CommandLine, SW_SHOWDEFAULT
    Invoke ExitProcess, eax
```

4.3.3 RadASM with UASM

The ModernUI x64 Library and ModernUI x64 Controls come with a RadASM project to help build the sources. To fully utilize this you may need to download and install:

- [UASM with RadASM](#)

and either:

- [UASM-SDK](#)

or

- [UASM](#)
- [WinInc](#)

- 64bit libraries - Can be obtained via: (assuming default installed locations)
 - Installed Windows SDK: \Program Files (x86)\Microsoft SDKs\Windows\v7.1A\Lib\x64
 - Installed Windows Kit: \Program Files (x86)\Windows Kits\8.1\Lib\winv6.3\um\x64
 - PellesC - \PellesC\Lib\Win64
- Other Binaries:
 - Resource Compiler: rc.exe, rcdll.dll
 - Resource Converter: cvtres.exe, cvtres.exe.config
 - Linker & Lib Manager: lib.exe, link.exe, link.exe.config, msobj120.dll, mspdb120.dll, mspdbcore.dll and the c runtime msvcrt120.dll

The UASM assembler and all related files (includes, libs, x64 libs, other binaries) should be placed in the appropriate folders so that your installation matches the following folder structure:

```
\UASM\bin
\UASM\include
\UASM\lib
\UASM\lib\x64
```

To add support for the UASM assembler to RadASM download and extract the [UASM with RadASM](#) package and edit the **RadASM.ini** file to add UASM32 and UASM64 to the Assembler entry under the Assembler section:

```
[Assembler]
Assembler=masm,UASM32,UASM64,JWasm,GoAsm,fasm,nasm,html
```

The RadASM projects for the ModernUI x64 Library and ModernUI x64 Controls should now assemble if all the above steps have been taken.

4.3.4 RadASM Colorized ModernUI Library Types

To add syntax color highlighting to the ModernUI library data types, open the RadASM assembler ini files: **masm.ini** and/or **uasm64.ini** and go to the following text section:

```
[KeyWords]
C6=
```

Append the following data types text to the C6 key in the KeyWords section:

```
^HDC ^HBITMAP ^HFONT ^HICON ^HANDLE ^HBRUSH ^HPEN ^MUIWND ^MUIPROPERTIES ^MUIPROPERTY
↪ ^MUIPROPERTYVALUE ^MUIVALUE ^LPMUIVALUE ^MUIIT ^MUIIL ^MUIPFS ^MUIIMAGE ^LPMUIIMAGE
↪ ^MUICOLORRGB ^MUICOLORARGB ^GPGRAPHICS ^LPGPGRAPHICS ^GPIMAGE ^LPGPIMAGE ^GPRECT ^
↪ LPGPRECT ^LPGDIPRECT ^LPRECT ^LPHBITMAP ^LPHDC ^POINTER ^RESID
```

4.4 Developers

The following details information on ModernUI development, for those wishing to understand the conventions used and using ModernUI and ModernUI controls as a user or a developer.

Functions/Procedures

- **Public** functions should be prefixed with the MUI (ModernUI) abbreviation
- **Private** functions should be prefixed with `_MUI_`

As a guideline, long code can be placed in their own procedure. Typically this might be for processing `WM_` messages (`WM_PAINT`) or where it helps readability.

Variables

Variables should be prefixed with Hungarian notation where possible

Control Libraries

- Control libraries should be prefixed with `ModernUI_` and then the name of the control.
- All controls should use there own include (.inc) file that has there own function prototypes, constants and global data variables (if applicable)
- The `ModernUI.lib` contains the main helper and utility functions for other MUI controls to use.
- The `ModernUI.inc` contains global constants used with the MUI framework.

Use of ModernUI

The inclusion of the `ModernUI.inc` and `ModernUI.lib` is required plus at a minimum one other control library and its own include file.

Control Properties (Users)

Properties can be changed (or read) by sending a custom message to the control via the standard `SendMessage` Win32 api call.

Typically properties (defined the .inc include file of the control) that can be changed will relate to colours (text, background, border), fonts and other various attributes of the control.

To get property values for a control the `MUI_GETPROPERTY` message is used, `wParam` is set to the property to obtain. To set property values for a control the `MUI_SETPROPERTY` message is used, `wParam` is set to the property to change, and `lParam` is set to the new property's value.

Example using an imaginary control that has a `@TextColor` property (defined in an include file) that allows the user to change the text color of the control:

```
Invoke SendMessage, hModernUIControl, MUISETPROPERTY, @TextColor, 00FFFFFFh
```

Controls will also provide their own Get / Set functions for convenience that achieve the same results as sending the `MUI_GETPROPERTY` / `MUI_SETPROPERTY` messages.

The list of properties that can be accessed or changed will be stored in the .inc include file of the control.

Control Properties (Developer Technical Notes)

The `cbWndExtra` field of the `WNDCLASSEX` Structure is used for storing pointers to two blocks of memory. One is for internal properties relating to the control, the second is for public exposed properties that can be set by the user. `cbWndExtra` should be set to 8 (16 for *ModernUI x64*) at a minimum to accommodate this when registering the new controls class.

Memory for both should be allocated during control creation (`WM_CREATE` or `WM_NCCREATE`) or after control creation, but before returning the control handle back to the user. The pointer to the block of memory for storing internal properties is stored at `cbWndExtra +0` and the pointer to the block of memory for storing external properties is stored at `cbWndExtra +4` (`cbWndExtra +8` for *ModernUI x64*)

The helper function `MUIAllocMemProperties` is provided to easily allocate the memory required and store it in the appropriate `cbWndExtra` offset. Here is an example from the `ModernUI_CaptionBar` (x86) control:

```
.ELSEIF eax == WM_CREATE
    Invoke _MUIAllocMemProperties, hWin, 0, SIZEOF _MUI_CAPTIONBAR_PROPERTIES ;_
↪internal properties
    Invoke _MUIAllocMemProperties, hWin, 4, SIZEOF MUI_CAPTIONBAR_PROPERTIES ;_
↪external properties
    Invoke _MUI_CaptionBarInit, hWin ; set some initial default property values
    mov eax, 0
    ret
```

Internal properties (to be changed by control developer) are defined as constant values which can be passed to the `_MUIGetIntProperty / _MUISetIntProperty` functions.

External properties (allowed to be changed by end-user) are defined as constant values which can be passed to the `MUIGetExtProperty / MUISetExtProperty` functions.

Controls should respond to the `MUI_GETPROPERTY / MUI_SETPROPERTY` messages, handling them with a forwarding call to `MUIGetExtProperty / MUISetExtProperty` functions and taking care of any other details as required.

Each control should also provide there own get / set property functions, which may just be wrappers for calls to the `MUI_GETPROPERTY / MUI_SETPROPERTY` messages. For controls that have other child controls that might be affected by a change to the parent's property value, this can be handled within `MUI_GETPROPERTY / MUI_SETPROPERTY` messages.

Each control should define there list of properties in their own include file for the user to reference.

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`