yesTRUEnono&AboutC&opy&PrintyesyesyesyesWILX Extender Help Filewilxyes28/02/01WILX Extender Help File

Table of Contents

Main

WILX EXTENDER HELP

Functions

xBaseConvert
Performs base conversions

xDiskLabelGet
Returns volume label of specified drive.

xDriveReady
Checks whether the drive is ready.

xExtenderInfo
Returns information on this extender.

xGetChildHwnd
Gets the handle of a child window belong to the specified parent.

xGetElapsed
Calculates difference between two values obtained with ExactTime().

xHex
Converts from hex to decimal.

xMessageBox
The "real" MessageBox function.

xSendMessage
The real "SendMessage".

xVerifyCCard
Verifies the validity of a credit card number.

xEjectMedia
Ejects media from removable drives.

For 16-bit Intel processors Extender DLL Name

AddExtender("wilx16i.dll")
Additional Dlls required: NONE

For 32-bit Intel processors Extender DLL Name

AddExtender("wilx34i.dll")
Additional Dlls required: NONE

This extender provides additional capability to the Windows Interface Language and is a good example of the useful things you as a developer can create using the Extender SDK.

Third party developers can customize and develop their own function libraries with Extender SDK. Custom extender Dlls may add nearly any sort of function to the WIL language, from the mundane network, math or database extensions, to items that can control fancy peripherals, including laboratory or manufacturing equipment. The Extender SDK can be purchased for $99.95. The source code to this extender, WILX is included as an example with the Extender SDK.

To use the Extender SDK you will need appropriate programming tools, such as Microsoft Visual C++ or the Borland Windows Development System.

xExtenderInfo(request #)

Returns information on this extender.

Syntax:

xExtenderInfo(request #)

Parameters:

(i) request # 0 = version
1 = total functions
2 = total constants

This function returns information on this extender. Version number, number of functions and number of constants.

Example:

```
AddExtender("wilx34i.dll")
ver = xExtenderInfo(0)
ftot = xExtenderInfo(1)
ctot = xExtenderInfo(2)
Message("WILX version %ver%", "Total functions: %ftot% %@CRLF% Total
                                              constants: %ctot%")
```

xBaseConvert(value, from-base, to-base)

Performs base conversions.

Syntax:

xBaseConvert(value, from-base, to-base)

Parameters:

(i)	value	number to convert (must be delimited if it contains letters.)
(i)	from-base	base that 'value' will be converted FROM
(i)	to-base	base that 'value' will be converted TO

This function performs base conversions.

Example:

```
AddExtender("wilx34i.dll")
dec = AskLine("Base conversion", "Enter a decimal value", 255)
hex = xBaseConvert(dec, 10, 16)
bin = xBaseConvert(dec, 10, 2)
Message("Base conversion", "Decimal: %@TAB% %dec% %@CRLF% Hex: %@TAB% %hex% %@CRLF%
Binary: %@TAB% %bin%")
hex = AskLine("Base conversion", "Enter a hex value", "D")
dec = xBaseConvert(hex, 16, 10)
bin = xBaseConvert(hex, 16, 2)
Message("Base conversion", "Decimal: %@TAB% %dec% %@CRLF% Hex: %@TAB% %hex% %@CRLF%
Binary: %@TAB% %bin%")
```

xHex(hex-val)

Converts from hex to decimal.

Syntax:

xHex(hex-val)

Parameters:

(i)	hex-val	hex number to convert (must be delimited if it contains letters)

Like xBaseConvert, but is a simpler way to perform the most useful conversion.

Example:

```
AddExtender("wilx34i.dll")
hex = AskLine("Hex to decimal conversion", "Enter a hex value", "FF")
dec = xHex(hex)
Message("Hex to decimal conversion", "Hex: %@TAB% %hex% %@CRLF% Decimal: %@TAB%
%dec%")
```

xGetElapsed(time1, time2)

Calculates difference between two values obtained with GetExactTime().

Syntax:

xGetElapsed(time1, time2)

Parameters:

(s)	time1	ending time
(s)	time2	start time

Example:

```
AddExtender("wilx34i.dll")
time1 = GetExactTime()
Display(1, "Elapsed time test", "Delaying for 1 second")
time2 = GetExactTime()
rc = xGetElapsed(time2, time1)
Message("Elapsed time", rc)
```

xMessageBox(title, text, style)

The "real" MessageBox function.

Syntax:
xMessageBox(title, text, style)

Parameters:
(s) title   title of message box
(s) text: text to go in message box
(s) style   type of icon and/or button(s) to use (the API documentation helps)

Returns the value assigned to the button that was pressed.

Style Parameter / Predefined Integer Constants

@MBOKCANCEL (= 1)
@MBYESNO (= 4)

Example:

```
AddExtender("wilx34i.dll")
mbstop     = xHex(10)
mbquestion = xHex(20)
mbexclaim  = xHex(30)
mbinfo     = xHex(40)
idyes      = xHex(6)
idno       = xHex(7)

rc = xMessageBox("MessageBox test", "Are you having fun?", @MBYESNO | mbquestion)

Switch rc
  case idyes
    xMessageBox("Good", "I'm glad to hear it", mbexclaim)
    break
  case idno
    xMessageBox("Sorry", "That's too bad", mbinfo)
    break
  case rc
    xMessageBox("Hey!", "We shouldn't be here!", mbstop)
    break
EndSwitch
```

xDiskLabelGet(drive)

Returns volume label of specified drive.

Syntax:

xDiskLabelGet(drive)

Parameters:

(s) drive	drive letter

Returns volume label of specified drive.

Example:

```
AddExtender("wilx34i.dll")
drives = DiskScan(6)
tot = ItemCount(drives, " ")
msg = ""

For i = 1 To tot
  drive = ItemExtract(i, drives, " ")
  label = xDiskLabelGet(drive)
  msg = StrCat(msg, "%drive%%@TAB%%label%%@CRLF%")
Next

Message("Volume labels", msg)
```

xEjectMedia(Drive)

Ejects media from removable drives.

Syntax:

xEjectMedia(drive)

Parameters:

(s) drive	drive letter

Sends a request to the hardware to eject the media cartridge. This function was designed for drives that can automatically unload the media cartridge, such as most CD-ROM and JAZ drives.

Example:

```
AddExtender("wilx34i.dll")
     drive = "s:"
     rc = xEjectMedia(drive)
     Message("xEjectMedia", rc)
```

xVerifyCCard(cardnum)

Verifies the validity of a credit card number.

Syntax:
xVerifyCCard(cardnum)

Parameters:

(i)	cardnum	credit card number, can include spaces or hyphens

Returns

1 if card number is valid, 0 if it isn't

This function verifies whether it is possible for a number to be a valid credit card number. It doesn't verify that the number is real, working credit card number. It only verifies that the number could be a valid credit card number.

Example:

```
AddExtender("wilx34i.dll")
card = AskLine("Credit card validation", "Enter a credit card number", "")
rc = xVerifyCCard(card)
If rc == 1 Then Message("Credit card number", "Is valid")
Else Message("Credit card number", "Is NOT valid")
```

xDriveReady(drive)

Checks whether the drive is ready.

Syntax:

xDriveReady(drive)

Parameters:

(s)	drive	drive letter

Checks whether the drive is ready indicating that there is a disk in the drive.
This is mainly useful for floppy and CD-ROM drives.

Example:

```
AddExtender("wilx34i.dll")
Pause("DriveReady test", "Put a floppy disk in Drive A:")
ready = xDriveReady("a:")
If ready Then Message("Drive A:", "Is ready")
Else Message("Drive A:", "Is NOT ready")

Pause("DriveReady test", "Now take the disk out of Drive A:")
ready = xDriveReady("a:")
If ready Then Message("Drive A:", "Is ready")
Else Message("Drive A:", "Is NOT ready")

Message("WILX", "The end")
```

xGetChildHwnd(parent-hwnd, child-text, child-seq)

Gets the handle of a child window belong to the specified parent.

Syntax:

xGetChildHwnd(parent-hwnd, child-text, child-seq)

Parameters:

(s)	parent-hwnd	handle of the parent window
(s)	child-text	text string of the child window (may be blank)
(i)	child-seq	position sequence of the child window (may be 0)

Returns:

(i)	the child handle or 0 if child window is not found.

Either child-text or child-seq, or both, must be specified.

If only the text is specified, it will be compared to all children until a match is
found; if not, 0 is returned.

If only the sequence is specified, it will be assumed to be correct; if there are less than 'seq' children, 0 is returned.

If both text and sequence are specified, the text is compared to only that one specific child; if they don't match, 0 is returned.

Text comparisons are case-insensitive, the '&' in the control string (if any) is stripped out, but leading and trailing spaces are not stripped.

Example:

```
; Determine whether 'Fast "Alt+Tab" Switching' checkbox in Control Panel
; "Desktop" dialog is checked, and then press the Cancel button
AddExtender("wilx34i.dll")

BMGETCHECK    = 1024
WMLBUTTONDOWN =  513
WMLBUTTONUP   =  514

If WinExist("Control Panel") == @FALSE Then Run("control.exe", "")
Then runpanel = 1
SendMenusTo("Control Panel", "Settings | Desktop")

hWnd = DllHwnd("Desktop")
Terminate(hWnd == 0, "Error", "Parent window not found")

; we know from experience that this control is sequence #6
hWndChild = xGetChildHwnd(hWnd, 'Fast "Alt+Tab" Switching', 6)
Terminate(hWndChild == 0, "Error", "Check box not found")
checked = xSendMessage(hWndChild, BMGETCHECK, 0, 0)

; on the other hand, we don't know the sequence # for the Cancel button
hWndChild = xGetChildHwnd(hWnd, "Cancel", 0)
Terminate(hWndChild == 0, "Error", "Cancel button not found")
xSendMessage(hWndChild, WMLBUTTONDOWN, 0, 0)
xSendMessage(hWndChild, WMLBUTTONUP, 0, 0)

If checked == 1 Then Message('Fast "Alt+Tab" Switching', "Box is checked")
Else Message('Fast "Alt+Tab" Switching', "Box is NOT checked")
```

```
If IsDefined(runpanel) Then WinClose("Control Panel")

:cancel
```

xSendMessage(hWnd, msg, wparam, lparam)

The real SendMessage.

Syntax:

xSendMessage(hWnd, msg, wparam, lparam)

Parameters:

(i) hWnd	window handle
(i) msg	message number
(i) wparam	first message parameter
(i) lparam	second message parameter (must be numeric)

Returns:

(i)	the value returned by the Windows "SendMessage" function.

This function is designed to allow sophisticated users to access the Windows API "SendMessage" function directly. **xSendMessage** calls the Windows API "SendMessage" function with the specified parameters and sends the specified message to the given window.

Example:

```
AddExtender("wilx34i.dll")
WMCLOSE = 16                                 ; value of WM_CLOSE message
AddExtender("wilx.dll")
hwnd = IntControl(21, "Notepad", 0, 0, 0)            ; get window handle
If hwnd != 0 Then xSendMessage(hwnd, WMCLOSE, 0, 0)  ; send message
```

See Also:

IntControl 22 *(WIL Reference Manual)*