

ListView

I note that Koda's ExStyles tab includes both WS_EX constants and LVS_EX constants. But there are LVS_EX constants that have the same value as WS_EX constants. For example: LVS_EX_REGIONAL and WS_EX_CLIENTEDGE, LVS_EX_UNDERLINECOLD and WS_EX_RIGHT.

MSDN, at <http://msdn.microsoft.com/en-us/library/windows/desktop/bb774732%28v=vs.85%29.aspx>, makes a point of saying that ListView Extended Styles should be set with the LVM_SETEXTENDEDLISTVIEWSTYLE – perhaps to avoid the conflicts.. Koda does not do this. I have found that some LVS_EX constants can be set in the normal way, i.e. in calling GuiCtrlListViewCreate. Others I have failed to get working both ways.

For LVS_REPORT, Melba23 has advised (on the GUI forum) that he understands that WS_EX constants belong in the GuiCtrlListViewCreate() call, and LVS_EX constants should be set by calling GuiCtrlSendMessage(-1, \$LVM_SETEXTENDEDLISTVIEWSTYLE, exStyle, exStyle). This is what my CreateTestControl() does.

Constants are set in my code as follows:

Report and List views

View	included in Style
LVS_ and WS_ styles	GuiCreateListView(... Style, ...)
LVS_EX styles	GuiSendMessage(-1, LVM_SETEXTENDEDLISTVIEWSTYLE, exStyle, exStyle)
WS_EX styles	GuiCreateListView(... ExStyle)

Icon and Small Icon views

View	included in Style
LVS_ and WS_ styles	GuiSetStyle(-1, Style)
LVS_EX styles	GuiSendMessage(-1, LVM_SETEXTENDEDLISTVIEWSTYLE, exStyle, exStyle)
WS_EX styles	GuiCreateListView(... ExStyle)

Tile view

View	GuiSendMessage(-1, LVM_SETVIEW, LV_VIEW_TILE)
LVS_ and WS_ styles	GuiSetStyle(-1, Style)
LVS_EX styles	GuiSendMessage(-1, LVM_SETEXTENDEDLISTVIEWSTYLE, exStyle, exStyle)
WS_EX styles	GuiCreateListView(... ExStyle)

\$GUI_BKCOLOR_LV_ALTERNATE is not a style as the following code shows:

```
#include <GUIConstantsEx.au3>
#include <GuiListView.au3>

GUICreate("AForm1", 370, 270, 193, 115)
$ListView1 = GUICtrlCreateListView("Name|Age|Sex", 10, 10, 350, 250)
GUICtrlSetBkColor($ListView1, $GUI_BKCOLOR_LV_ALTERNATE)
GUICtrlCreateListViewItem("Joe|26|Male", $ListView1)
```

```

GUICtrlCreateListItem("Mary|48|Female", $ListView1)
GUICtrlSetBkColor(-1, 0xf3f389)
GUICtrlCreateListItem("Jack|6|Male", $ListView1)
GUICtrlCreateListItem("Kittie|99|Female", $ListView1)
GUICtrlSetBkColor(-1, 0xf3f389)
GUISetState(@SW_SHOW)

```

```

while 1
    $nMsg = GUIGetMsg()
    Switch $nMsg
        Case $GUI_EVENT_CLOSE
            Exit
    EndSwitch
wend

```

so \$GUI_BKCOLOR_LV_ALTERNATE is omitted.

LVS_ALIGNMASK = 0x0c00, so alignment includes both LVS_ALIGNLEFT and LVS_OWNERDRAWFIXED. Per MSDN, LVS_ALIGNLEFT only applies to Icon and Small Icon views but LVS_OWNERDRAW only applies to Report view. I am confused! I have included LVS_OWNERDRAWFIXED for now.

I have doubts as to whether LVS_EX_UNDERLINECOLD works. Hovering to underline only works with LVS_EX_UNDERLINEHOT when calling UDFs rather than AutoIt's GuiCtrlCreateListView functions.

In Report view, combinations of LVS_EX_ONECLICKACTIVATE / LVS_TWOCLICKACTIVATE and LVS_EX_UNDERLINEHOT / LVS_EX_UNDERLINECOLD with AutoIt's GuiCtrlCreateListView() and all UDF ListView functions:

Click Const	Underline Const	Initial State	Hovering underlines	First click	Second click
ONE	none	normal	no	selects and notifies	not applicable
ONE	HOT	normal	yes	selects and notifies	not applicable
ONE	COLD	all underlined	no	selects and notifies	not applicable
TWO	none	normal	no	underlines	selects and notifies
TWO	HOT	normal	no	underlines	selects and notifies
TWO	COLD	normal	no	underlines	selects and notifies
none	HOT	normal	no	highlights	selects*
none	COLD	normal	no	highlights	selects*

* double-clicking notifies

MSDN's documentation is confusing, so the dialog allows all combinations. MSDN does say that:

1. LVS_EX_UNDERLINECOLD requires LVS_TWOCLICKACTIVATE
2. LVS_EX_UNDERLINEHOT requires either LVS_EX_ONECLICKACTIVATE or LVS_TWOCLICKACTIVATE

WS_CLIPSIBLINGS only works in Icon and Small Icon view, so it is disabled in other views. The control that has this style must be created after the one with which it overlaps.

Icon and Small Icon views do not arrange the icons properly if the control is too small; hence the larger test form.

Tile view functions were found at http://code.google.com/p/m-a-t/downloads/detail?name=GUIListView_Additions.zip . These functions are intended to be UDFs. It seems that even with these functions, at least on XP, AutoIt can only do fairly crude Tile view. I was unable to find any information on the Internet, I assumed that Tile view has the same styles as Report, plus `LVS_EX_COLUMNOVERFLOW`.

Which styles apply to which views is per MSDN. While MSDN seems to be generally accurate for ListView, I have found that `LVS_EX_BORDERSELECT` only seems to work for Icon view. Also, this style highlights the text as well as colouring the border: MSDN say it only colours the border.