

The NTK Project
for
xHarbour

NtkTBrowse CLASS
Programmer's Guide

©2001-2006 - Jn Dechereux.

Summary

I	Introduction	1
0.1	What is NtkTBrowse	3
0.2	Overview	3
II	NtkColumn Class	5
0.3	Method	7
0.3.1	NtkColumn:Init()	7
0.4	Properties and Instance variables	11
0.4.1	Heading	11
0.4.2	Block	12
0.4.3	Picture	13
0.4.4	fgColor	14
0.4.5	BgColor	15
0.4.6	Align	16
0.4.7	Width	17
0.4.8	Bitmap	18
0.4.9	BitAlign	19
0.4.10	HeadBlock	20
0.4.11	Cargo	21
0.4.12	Font	22
0.4.13	HeadColor	23
0.4.14	HeadFont	24
0.4.15	HeadAlign	25
0.4.16	HeadBmpIndex	26
0.4.17	HeadBmpAlign	27
0.4.18	HeadBmpVAlign	28

III	NtkTBrowse Class	29
0.5	Methods	31
0.5.1	Init()	31
0.5.2	AddColumn()	34
0.5.3	Bottom()	35
0.5.4	Configure()	36
0.5.5	DelColumn()	37
0.5.6	FindColumn()	38
0.5.7	Destroy()	39
0.5.8	Down()	40
0.5.9	End()	41
0.5.10	GetCellData()	42
0.5.11	GetCellRect()	43
0.5.12	GetColCargo()	44
0.5.13	GetColumn()	45
0.5.14	GetNubData()	46
0.5.15	GoBottom()	47
0.5.16	GoTop()	48
0.5.17	Home()	49
0.5.18	InsColumn()	50
0.5.19	Left()	51
0.5.20	MakeColVisible()	52
0.5.21	ModColumn()	53
0.5.22	MoveColumn()	54
0.5.23	PageDown()	55
0.5.24	PageUp()	56
0.5.25	PanEnd()	57
0.5.26	PanHome()	58
0.5.27	PanLeft()	59
0.5.28	PanRight()	60
0.5.29	RefreshAll()	61
0.5.30	RefreshCurrent()	62
0.5.31	Right()	63
0.5.32	SetColWidth()	64
0.5.33	SetRowHeight()	65
0.5.34	ShowAll()	66
0.5.35	ShowCurrent()	67
0.5.36	ShowData()	68
0.5.37	StandByMode()	69
0.5.38	Top()	70

0.5.39	Up()	71
0.6	Properties and Instance variables	72
0.6.1	alias	72
0.6.2	autoLite	73
0.6.3	autoSize	74
0.6.4	beeper	75
0.6.5	boxCursor	76
0.6.6	cargo	77
0.6.7	colCount	78
0.6.8	colFont	79
0.6.9	colGridColor	80
0.6.10	colGrids	81
0.6.11	colPos	82
0.6.12	colorSpec	83
0.6.13	destroyBlock	84
0.6.14	doubleclick	85
0.6.15	drawingCol	86
0.6.16	drawingRow	87
0.6.17	escape	88
0.6.18	headFont	89
0.6.19	headColor	90
0.6.20	headHeight	91
0.6.21	hitBottom	92
0.6.22	hitTop	93
0.6.23	hHeader	94
0.6.24	hNubWnd	95
0.6.25	hWnd	96
0.6.26	keyBlock	97
0.6.27	leftVisible	98
0.6.28	lineHeight	99
0.6.29	make3D	100
0.6.30	mimicButton	101
0.6.31	msgBlock	102
0.6.32	nCrecNo	103
0.6.33	nCurRec	104
0.6.34	nMaxRec	105
0.6.35	nubAlign	106
0.6.36	nubBlock	107
0.6.37	nubHeadBlock	108
0.6.38	nubHheadBlock	109

0.6.39	nubHBitmap	110
0.6.40	nubHBitAlign	111
0.6.41	nubClickBlock	112
0.6.42	nubColor	113
0.6.43	nubHeading	114
0.6.44	nubHeadClickPos	115
0.6.45	nubHeadColor	116
0.6.46	nubWidth	117
0.6.47	onGainFocus	118
0.6.48	onLooseFocus	119
0.6.49	rightClick	120
0.6.50	rightVisible	121
0.6.51	rowCount	122
0.6.52	rowGridColor	123
0.6.53	rowGrids	124
0.6.54	rowPos	125
0.6.55	saveColSize	126
0.6.56	sbHorz	127
0.6.57	sbVert	128
0.6.58	selectblock	129
0.6.59	showFocus	130
0.6.60	showNumbers	131
0.6.61	singleClick	132
0.6.62	tbFont	133
0.6.63	userMove	134
0.6.64	userSize	135
0.7	Related functions.	136
0.7.1	NtkTBGetCurrent(void)	136
IV	Miscellaneous.	137
0.8	Picture and Bitmap support.	139
0.9	Default navigation keys and mouse behaviour.	140
0.10	NtkTBrowse restrictions.	141

Part I

Introduction

0.1 What is NtkTBrowse

NtkTBrowse is a GUI customizable DataGrid object class, based upon the Win32 Listview API of NTKCore, it is very similar in its way of programming to the console Tbrowse xHarbour (by the way with CA-Clipper) and is also mostly backward compatible with the C4W's WBrowse class.

0.2 Overview

A NtkTBrowse main purpose is to generate a browsing mechanism for table-oriented data. NtkTBrowse objects provide all the needed architecture to accurately get, format, and display datas. Datas retrieval and file positioning are performed via the NtkTBrowse's internal code block. This block ensure the correct interaction between the browsing and drawing mechanism, and the underlying data source. The format of individual data items can be precisely controlled via the NTKColumn data retrieval code blocks; overall display formatting and attributes can be controlled by sending appropriate messages to the NtkTBrowse object.

A NtkTBrowse object relies on one or more NTKColumn objects. A NTKColumn object contains the information necessary to define a single column of the browse table (see NTKColumn class in the following sections).

During operation, a NtkTBrowse object retrieves datas by evaluating code blocks. Datas are organized into rows and columns and displayed within a window parent container. The TBrowse object maintains an internal browse cursor. The data item on which the browse cursor rests is displayed in a highlighted color.

Initially, the browse cursor is placed on the data item at the top left of the browse display. Messages can then be sent to the NtkTBrowse object to navigate the displayed data, causing the browse cursor to move. These messages are normally sent in response to user keystrokes, mousewheel interaction or other GUI controls (i.e. sliders, buttons, etc...).

New data is automatically retrieved as required by navigation requests. When navigation proceeds past the edge of the visible rectangle, rows or columns beyond that edge are automatically brought into view. When new rows are brought into view, the underlying data source is repositioned by evaluating NtkTBrowse's internal code block.

Examples

See:

`\WNTK4HRB\NTKRAD\BRWIMAGE.prg`

`\WNTK4HRB\NTKRAD\BRTEST.prg`

Part II

NtkColumn Class

0.3 Method

0.3.1 NtkColumn:Init()

NTKColumn:init() Creates a new NTKColumn object.

Syntax

```
NtkColumn:Init( [<cheading>], [<bBlock>], [<cpicture>]           ;;
                [<fgcolor>], [<bgcolor>]                       ;;
                [<nalign>], [<nWidth>]                         ;;
                [<ncBitmap>], [<nBitAlign>]                   ;;
                [<bHeadblock>], [<xCargo>], [<hFont>]          ;;
                [<nHeadFgColor>], [<hHeadFont>], [<nHeaderAlign>], ;;
                [<nBmpheaderID>]                               ;;
                [<nBmpHeaderAlign>], [<nBmpHeaderVAlign>] ) -> NTKColumn Object
```

Description

This method creates a new NTKColumn object and returns a new NTKColumn object with the specified heading and data retrieval block. Other elements of the NTKColumn object can be assigned directly using the syntax for assigning exported instance variables.

Arguments

<heading> - Heading title for the column.

It is possible to make a multi-line headers using one of following ways.

- a - By passing a mono dimensional array containing a string for each line.
- b - By passing strings representing each line, but CR (Chr(13)) separated.

<bBlock> - Data retrieval code block for the column. Multi-line data text is available only for character fields. Each field must be separated with a CR (Chr(13)).

<cpicture> - Optional picture clause. Must conform to the TRANSFORM() function.

<fgcolor> - Optional code block that returns a valid NTK_RGB() value for the foreground color of the data cell to be displayed. Default is the value specified in ColorSpec for the un-selected foreground color.

<bgcolor> - Optional code block that returns a valid NTK_RGB() value for the background color for the data cell to be displayed. Defaults to the value specified in ColorSpec for the un-selected background color.

<nAlign> - Optional value that specifies how the data text will be aligned within the column. Valid values are DT_LEFT, DT_CENTER, and DT_RIGHT. Defaults to DT_LEFT.

<nWidth> - Optional column width (in pixels).

<ncBitmap> - Optional. It can be the numeric ID of the bitmap resource to be used in the column data cells. In this case, the bitmap MUST be included in your resource file. Or, it can also be a string value representing the complete pathfolder and name of the picture to be displayed. (i.e "C:\MyAlbum\mypict.jpg")

<nbitalign> - Optional horizontal alignment for column heading bitmap. Use DT_LEFT, DT_CENTER, or DT_RIGHT. Defaults to DT_CENTER.

Combination with vertical alignment flags such as DT_TOP, DT_VCENTER, or DT_BOTTOM are now supported since version 1.0 of NTK.

<bHeadblock> - Optional codeblock to be evaluated when the user clicks on this column's heading.

<xCargo> - Optional cargo for the column. This can be any data type.

<hFont> - Optional font for the column data cells. Must be a handle to a valid Windows font or a stockobject.

<nHeadfgColor> - Optional color value to use when drawing the column text heading. Must be a valid value returned from NTK_RGB().

<hHeadFont> - Optional font for the column text heading. Must be a handle to a valid Windows font or a stockobject.

<nHeaderAlign> - Optional value that specifies how the text will be aligned within the column header. Valid values are DT_LEFT, DT_CENTER, and DT_RIGHT. Defaults to DT_LEFT.

<nBmpHeaderID> - Optional. It can be the numeric ID of the bitmap resource to be used in the column header. The bitmap MUST be included in your resource file. Or, it can also be a window stockobject defined by an OBM_* value.

<nBmpHeaderAlign> - Optional value that specifies how the bitmap will be horizontally aligned within the column header. Valid values are DT_LEFT, DT_CENTER, and DT_RIGHT. Defaults to DT_LEFT.

<nBmpHeaderVAlign> - Optional value that specifies how the bitmap will be vertically aligned within the column header. Valid values are DT_LEFT, DT_CENTER, and DT_RIGHT. Defaults to DT_LEFT.

Notes:

All parameters are optional. When no parameter is passed an empty column object is returned. You can then assign values to the column objects properties individually.

It is recommended that code block for each column returns a character type unless the picture parameter is used. In the same way, Dates should be converted using DTOC() or DTOS() and Memo fields should be avoided, because of their possibility of big size and the resulting impact on browse performance!

If you want to set the column sizes yourself take care to set autoSize to .F. or saveColSize to .T.

You can use bitmaps in a column, blent to normal data text. When using bitmaps the column block MUST return a numeric value equal to the ID of the bitmap in your resource file, or 0, or an OBM_* value representing a window internal bitmap/stockobject)

Examples

```
// Create a new column object...
oCol1 := NtkColumn():Init( "Purchase_price",;
                          { || STR(PRODUCT->PUAHT,10,2) },;
                          nil, nil, nil, DT_RIGHT, 150 )

// ----- #1 Column : Optional Customization...
oCol1:HeadColor      := NTK_RGB(0,128,0) // Title will appear in Dark Green
oCol1:HeadFont       := NTK_GetStockObject(DEFAULT_GUI_FONT)
oCol1:HeadBmpIndex   := 1501
oCol1:HeadBmpAlign   := DT_CENTER
oCol1:HeadBmpVAlign  := DT_TOP
oCol1:fgColor        := {|| IIF(PRODUCT->PUAHT<1,NTK_RGB(255,0,0), Nil) }
```


0.4 Properties and Instance variables

0.4.1 Heading

Char or Array value. (Access/Assign)

Heading title for the column.

It is possible to make a multi-line headers using one of following ways.

- a - By passing a mono dimensional array containing a string for each line.
- b - By passing strings representing each line, but CR (Chr(13)) separated.

0.4.2 Block

Block value. (Access/Assign)

Data retrieval code block for the column. Multi-line data text is available only for character fields. Each field must be separated with a CR (Chr(13)).

0.4.3 Picture

Char value. (Access/Assign)

A string containing a valid picture conform to the TRANSFORM() function.

Example:

```
oCol:picture := "99,999.99" // Format numbers this way...
```

0.4.4 fgColor

Numeric value. (Access/Assign)

RGB value or code block that returns a valid `NTK_RGB()` value for the foreground color of the data cell to be displayed. Default is the value specified in `ColorSpec` for the un-selected foreground color.

0.4.5 BgColor

Numeric value. (Access/Assign)

RGB value or code block that returns a valid NTK.RGB() value for the background color for the data cell to be displayed. Defaults to the value specified in ColorSpec for the un-selected background color.

0.4.6 Align

Numeric value. (Access/Assign)

Value that specifies how the data text will be aligned within the column. Valid values are DT_LEFT, DT_CENTER, and DT_RIGHT. Defaults to DT_LEFT.

0.4.7 Width

Numeric value. (Access/Assign)

Column size (in pixels).

0.4.8 Bitmap

Numeric or Char value. (Access/Assign)

It can be the numeric ID of the bitmap resource to be used in the column data cells. In this case, the bitmap MUST be included in your resource file. Or, it can also be a string value representing the complete pathfolder and name of the picture to be displayed.
(i.e. "C:\MyAlbum\mypict.jpg")

0.4.9 BitAlign

Numeric value. (Access/Assign)

Alignment for column heading bitmap. Use DT_LEFT, DT_CENTER, or DT_RIGHT. Defaults to DT_CENTER. Combination with vertical alignment flags such as DT_TOP, DT_VCENTER, or DT_BOTTOM are now supported since version 1.0 of NTK.

0.4.10 HeadBlock

Block value. (Access/Assign)

A code block to be evaluated when the user clicks on this column's heading.

0.4.11 Cargo

Value of any type. (Access/Assign)

Cargo for the column. This can be any data type.

0.4.12 Font

Numeric value. (Access/Assign)

Font for the column data cells. Must be a handle to a valid Windows font or a stockobject.

0.4.13 HeadColor

Numeric value. (Access/Assign)

Color value to use when drawing the column text heading. Must be a valid value returned from NTK_RGB().

0.4.14 HeadFont

Numeric value. (Access/Assign)

Font for the column text heading. Must be a handle to a valid Windows font or a stockobject.

0.4.15 HeadAlign

Numeric value. (Access/Assign)

Value that specifies how the text will be aligned within the column header. Valid values are DT_LEFT, DT_CENTER, and DT_RIGHT. Defaults to DT_LEFT.

0.4.16 HeadBmpIndex

Numeric value. (Access/Assign)

It can be the numeric ID of the bitmap resource to be used in the column header. The bitmap **MUST** be included in your resource file. Or, it can also be a window stockobject defined by an OBM_* value.

0.4.17 HeadBmpAlign

Numeric value. (Access/Assign)

Value that specifies how the bitmap will be horizontally aligned within the column header. Valid values are DT_LEFT, DT_CENTER, and DT_RIGHT. Defaults to DT_LEFT.

0.4.18 HeadBmpVAlign

Numeric value. (Access/Assign)

Value that specifies how the bitmap will be vertically aligned within the column header. Valid values are DT_LEFT, DT_CENTER, and DT_RIGHT. Defaults to DT_LEFT.

Part III

NtkTBrowse Class

0.5 Methods

0.5.1 Init()

NtkTBrowse():Init() Creates a new browse object into memory.

Syntax

```
NtkTBrowse():Init( <hWnd>,;
                  <nLeft>, <nTop>, <nLarg>, <nHaut>,;
                  [<cTitle>], [<nStyle>] ) -> TBrObj
```

Description

This method creates a new NtkTBrowse object as a new child control of a parent window's container, and returns a new NtkTBrowse object with the specified features. Then, columns previously initialized using NtkColumn:Init() MUST be connected using NtkTbrowse:AddColumn() method.

Arguments

<hWnd> - Handle to parent window

<nLeft> - Column number (in pixels) of top left corner of browse window

<nTop> - Row number (in pixels) of top left corner of browse window

<nLarg> - Width in pixels of browse window

<nHaut> - Height in pixels of browse window

<cTitle> - Optional title for browse window

<nStyle> - Optional style for browse window.

Must be a valid combination of WS_* values used in the NTK_CreateWindow() function call.

Default is set to:

```
WS_CAPTION+WS_SYSMENU+WS_MINIMIZEBOX+WS_THICKFRAME+WS_MAXIMIZEBOX+WS_VSCROLL+WS_HSCROLL.
```

Examples

```
// #1 Column : Creation...
```

```

oCol1:=NtkColumn():Init( "Picture" , { || dbImage->PICTNAME } , nil, nil, nil,

// ----- #1 Column : Optional Customization...
// Column NubHeading aspect
oCol1:HeadAlign      := DT_CENTER+DT_VCENTER
oCol1:HeadBmpIndex   := nil           // Bitmap res to use in the NubHeading or OBM_*
oCol1:HeadBmpAlign   := DT_CENTER
oCol1:HeadBmpVAlign  := DT_TOP
// Bitmap aspect inside cells.
// Can be a numeric Bitmap Res ID or OBM_*, or a char PictureName from hdd
oCol1:Bitmap         := { || IIF( EMPTY(dbImage->PICTNAME), 0, cExplorePath+ALLTRIM(dbI
oCol1:BitAlign:= DT_CENTER+DT_VCENTER // DT_LEFT+DT_TOP

// Create browse object into memory
oB := NtkTBrowse():Init( hWndMain ,;
                        010,060 , 160, 680,; //__NTKMaxRow()-060-40-100,;
                        "",;
                        WS_CHILD + WS_VISIBLE + WS_OVERLAPPED +WS_THICKFRAME+ W

// Connect Column(s) to browse object
oB:addColumn( oCol1 )

// ----- Browse Settings (aspect, behaviour)
oB:AutoLite          := .F. // .F. Hilight focused cell / .T. Hilight all cells of the
oB:boxcursor         := .T. // Mimics SpreadSheet Cursor focus. (Black thin borders)
oB:make3D            := .F. // 3D look : Each cell appears to be raised
oB:mimicButton:= .F. // Vertical & heading Nubs will work like buttons
oB:ColGrids          := .T. // We Want LT_GRAY lines (Grid) between each row/col
oB:RowGrids          := .T. // We Want LT_GRAY lines (Grid) between each row/col
oB:userMove          := .F. // Allows Col Drag & Drop
oB:userSize          := .F. // Allows Col Resize
oB:escape            := .F. // User can/can't exit the browse pressing Esc Key. Defaut

oB:HeadHeight       := 30
**oB:Headfont        := NTK_GetStockObject(DEFAULT_GUI_FONT) //ANSI_VAR_FONT ANSI_FI
oB:HeadColor         := NTK_RGB( 0,0,255 )
oB:tbfont            := NTK_GetStockObject(DEFAULT_GUI_FONT) // ANSI_FIXED_FONT
oB:LineHeight        := 100 // 20 // Set the height of the browse rows in pixel.

```

```
oB:ColorSpec := { NTK_RGB( 0, 0, 255) , NTK_RGB(255, 255, 176) ,; // UnSel C
                  NTK_RGB(255, 255, 255) , NTK_RGB(156, 131, 247) ,; // Sel Cel
                  NTK_RGB(184, 184, 184) ; // WndBrow
                }

// Browse object Configuration and activation
oB:Configure()
NTK_SetFocus(oB:hWnd)

//
// Ok, here we go...
//
```

0.5.2 AddColumn()

Adds a new NtkTBColumn object to the NtkTBrowse object and NtkTBrowse:colCount is increased by one.

Example:

```
oCol1:=NtkColumn():Init( "Part_number",;
                        { || PRODUCT->CODEART },;
                        "@!",;
                        NTK_RGB(255,255,228),;
                        NTK_RGB(55,240,23),;
                        DT_RIGHT,;
                        100 )

// ----- #1 Column : Optional Customization...
oCol1:HeadColor      := NTK_RGB(0,128,0) // Title will appear in Dark Green
oCol1:HeadFont       := NTK_GetStockObject(DEFAULT_GUI_FONT)
oCol1:HeadBmpIndex   := 1563 // My Resource ID: a codbar bitmap
oCol1:HeadBmpAlign   := DT_RIGHT
oCol1:HeadBmpVAlign  := DT_BOTTOM

oB:addColumn( oCol1 )

// etc, etc...
```

0.5.3 Bottom()

Moves the browse record pointer to the bottom of the database being browsed.

0.5.4 Configure()

Causes the TBrowse object to reexamine all instance variables and NtTBColumn objects, reconfiguring its internal settings as required.

This message can force reconfiguration when a NtkTBColumn object is modified directly.

Example:

```
// Delete column #3 ...  
oB:DelColumn( 3 )  
oB:configure()  
oB:refreshAll()
```

0.5.5 DelColumn()

DelColumn(<nCol>) -> oCol

Allows a specified column object to be removed from the browse.

The return value is a reference to the column object before to be deleted, or Nil if it fails.

Note that after deleting, you MUST call NtkTBrowse:Configure(), then NtkTBrowse:RefreshAll() to update the browse.

Example:

```

FUNCTION DelCol( oB )
LOCAL nSelCol := oB:ColPos
LOCAL cMsg    := "Do you really want to remove column #"+
                LTRIM(STR(nSelCol))+"? "

IF NTK_MsgBox( oB:hWnd, cMsg, "ATTENTION!!!",;
              MB_OKCANCEL+MB_ICONQUESTION ) == IDOK
  // Delete column # ...
  oB:DelColumn( nSelCol )
  oB:configure()
  oB:refreshAll()
ENDIF
RETURN(nil)

```

0.5.6 FindColumn()

FindColumn(<cColTitle>, <lStrictSearch>) -> nCol

Retrieves the order position of a column object within the browse by using its heading string.

<cColTitle> is the heading string of the column to find.

<lStrictSearch> Optional. Default is always .T.

If this parameter is disable (.F.) the research algorithm doesn't compare on a strict equality of the two strings, but just ensures that cColTitle is found inside the heading string of the scanned column. The first match is returned.

The return value is the index order (position) of the found column object within the browse, or Zero if it fails.

Example:

```
nColFound := oB:FindColumn( "Product_Description" )
IF nColFound > 0
    oCol      := oB:GetColumn(nColFound)

    oNewCol := oCol
    oNewCol:HeadBmpIndex := OBM_CHECK // OBM_CHECK, OBM_COMBO
    oNewCol:HeadBmpAlign := DT_RIGHT
    oNewCol:HeadBmpVAlign := DT_BOTTOM // DT_TOP, DT_BOTTOM, DT_CENTER

    oB:ModColumn( nColFound, oNewCol )
ENDIF
```

0.5.7 Destroy()

Kill the current browse object.

Please note that the Destroy method is automatically called when the browse window is closed or sent a WM_DESTROY message.

Syntax:

Destroy() -> Self

Arguments:

None.

Return:

Self.

Example:

```
oB:Destroy() // Kill browse object: oB
```

40

0.5.8 Down()

0.5.9 End()

0.5.10 GetCellData()

Retrieves data contents of the current or specified cell.

Syntax:

```
GetCellData([<nRow>],[<nCol>]) -> aCellData
```

Arguments:

nRow: optional row position of the cell to get data

nCol: optional Col position of the cell to get data

Return:

An array containing both data and picture (if any) of the designed cell: data, picture.

Example:

```
aCellData := oB:getCellData() // getCellData(nRow,nCol) -> {xData , Picture}  
xCellValue := aCellData[1]
```

0.5.11 GetCellRect()

Retrieves coordinates of the current or specified cell.

Syntax:

```
GetCellRect([<nRow>],[<nCol>]) -> aCellRect
```

Arguments:

nRow: optional row position of the cell to get data

nCol: optional Col position of the cell to get data

Return:

An array containing coordinates and sizes of the selected or designed cell: nLeft, nTop, nWidth, nHeight.

Example:

```
// getCellRect(nRow,nCol) -> {Left,Top,Width,Height}
aCellRect := oB:getCellRect(oB:RowPos, oB:ColPos)
nCellY     := aCellRect[RECT_Top]      +1
nCellX     := aCellRect[RECT_Left]     +1
nCellWidth := aCellRect[RECT_Right]    -1
nCellheight := aCellRect[RECT_Bottom]  -2
```

0.5.12 GetColCargo()

Retreives a column cargo value previoully set by the developer.

Syntax:

GetColCargo(<nCol>) -> xCargoValue

Arguments:

nCol: Column number from which to get the cargo data.

Return:

A value of any type depending of what the developer puts in it... Nil is returned if an error occurs or a wrong column number has been passed.

Example:

0.5.13 GetColumn()

Retreives a NtkTBrowse column object.

Syntax:

GetColumn(<nCol>) -> objColumn

Arguments:

nCol: number of the Column to get.

Return:

A NtkTBcolumn object for the specified column, or Nil if <nCol> is invalid.

Example:

```
FOR nC := 1 TO oB:ColCount
  oCol := oB:GetColumn( nC )
  AADD( aCols, oCol )
NEXT
```

0.5.14 GetNubData()

Retreives data content of the current or specified rowNum.

Syntax:

```
GetNubData([<nRow>]) -> xNubData
```

Arguments:

nRow: optional row position of the vertical nub to get data

Return:

The content (of any type) of the designed vertical nub. This result is depending of the evaluation of the code-block specified into the 'nubBlock' instance var.

Example:

```
xNubValue := oB:getNubData(oB:DrawingRow) // Here we get nubdata of the current pa
```

0.5.15 GoBottom()

48

0.5.16 GoTop()

0.5.17 Home()

0.5.18 InsColumn()

InsColumn(<nPos>, <oCol>) -> oCol

Allows a column object to be inserted within a browse, at nPos.

The return value is a reference to the column object being inserted, or Nil if it fails.

Note that after inserting, you MUST call NtkTBrowse:Configure(), then NtkTBrowse:RefreshAll() to update the browse.

Example:

```

FUNCTION InsNewCol( oB )
LOCAL oNewCol
LOCAL nSelCol := oB:nubHeadClickPos

// Add new column ...
oNewCol := NtkColumn():Init( "Stock" ,;
                             { || STR(PRODUCT->STOCTACTUEL,10,3) },;
                             nil,;
                             NTK_RGB(12,15,46),;
                             NTK_RGB(255,255,255),;
                             DT_RIGHT,;
                             50 )

// Heading's text will appear in purple color...
oNewCol:HeadColor := NTK_RGB(255,255,128)

oB:insColumn( nSelCol, oNewCol )
oB:configure()
oB:refreshAll()

NTK_SetFocus(oB:hWnd)
RETURN(nil)

```

0.5.19 Left()

Move the cursor left one column.

0.5.20 MakeColVisible()

Ensure the specified column to be visible within the browse area.
This method can also be used to move the browse to the specified nCol.

Syntax:

MakeColVisible(<nCol>) -> self

Arguments:

nCol: Is the number of the column to force to be visible

Return:

Self.

Example:

```
oB: MakeColVisible( oB: ColPos+3 )  
oB: ShowCurrent()
```

0.5.21 ModColumn()

Modify properties of a column 'on fly' in an active browse.

Syntax:

ModColumn(<nCol>, <oNewCol>) -> Self or Nil

Arguments:

<nCol>: Index of the column to modify.

<oNewCol>: a valid Column object with the new column properties.

Return:

Result is Self if nCol and oNewCol are both valid, otherwise Nil is returned.

Example:

```
oOldCol := oB:GetColumn( 3 )

oNewCol := oOldCol
oNewCol:Font      := ANSI_VAR_FONT           // Set small font
oNewCol:fgcolor   := {||RGB(255, 0, 0) } // Make data red
oNewCol:bgcolor   := {||RGB(255, 255, 255) } // on white...

oB:ModColumn( 3, oNewCol )
oB:Configure() // Must be called right now!
oB:refreshAll() // Not very useful, just for C4W backward compatibility.
```

0.5.22 MoveColumn()

Move a column from its original place to a destination place.

Syntax:

MoveColumn(<nSrcCol>, <nDstCol>) -> IRet

Arguments:

<nSrcCol>: Index of the column to move.

<nDstCol>: Index of destination column where it has to be moved.

Return:

Result is .T. if the column has been successfully moved, otherwise it's .F.

Example:

```
oB::MoveColumn( nColToMove , nColToMove+4 )  
oB::Configure() // Must be called right now!
```

0.5.23 PageDown()

Move the record pointer down one page

56

0.5.24 PageUp()

Move the record pointer up one page

0.5.25 PanEnd()

Move the cursor to first column of the browse.

0.5.26 PanHome()

Move the cursor to the last column of the browse.

0.5.27 PanLeft()

Move the cursor left one page.

60

0.5.28 PanRight()

Move the cursor right one page.

0.5.29 RefreshAll()

Causes a complete reloading of data rows and redraws the whole browse window.

Syntax:

RefreshAll() -> Self

Arguments:

None.

Return:

Self.

Example:

```
// Update browse  
oB.RefreshAll()
```

0.5.30 RefreshCurrent()

Reloads and redraws the datas of current record only.

Syntax:

RefreshCurrent() -> Self

Arguments:

None.

Return:

Self.

Example:

```
// Update cells of current row  
oB.RefreshCurrent()
```

0.5.31 Right()

Move the cursor right one column.

0.5.32 SetColWidth()

Changes the size of a column (in pixel).

Please note that If you use this function you should set autoSize to .F. - You also must call configure() and refreshAll() afterwards.

Syntax:

SetColWidth(<nCol>, <nWidth>) -> Self

Arguments:

nCol: Column number.

nwidth: New width to apply to the Column.

Return:

Self.

Example:

```
// Change the width of column #1...
oB:SetColWidth( 1, 120 )
oB:Configure()
oB:RefreshAll()
```

0.5.33 SetRowHeight()

Changes browse row height (in pixel).

Please note that If you use this function you should set autoSize to .F. - You also must call configure() and refreshAll() afterwards.

See also the **lineHeight** instance variable.

Syntax:

SetRowHeight(<nHeight>) -> Self

Arguments:

nHeight: New Height to apply to all lines of the browse.

Return:

Self.

Example:

```
// Change browse rows to 22 pixels height.  
oB:SetRowHeight( 22 )  
oB:Configure()  
oB:RefreshAll()
```

0.5.34 ShowAll()

Just redraws the whole browse window without reloading data rows.

Syntax:

ShowAll() -> Self

Arguments:

None.

Return:

Self.

Example:

```
oB : ShowAll ()
```

0.5.35 ShowCurrent()

Redraws all cells of the current line.

Syntax:

ShowCurrent() -> Self

Arguments:

None.

Return:

Self.

Example:

```
// Update cells of current row  
oB: ShowCurrent()
```

0.5.36 ShowData()

Same as refreshAll(), but headings and browse background are not repainted. Much faster...

Syntax:

showData() -> Self

Arguments:

None.

Return:

Self.

Example:

```
// Update browse  
oB: showData()
```

0.5.37 StandByMode()

Enables or disables browse 'frozen' mode.

While Standbymode is active, no operation or motion are possible within the browse.

Syntax:

StandByMode(IMode) -> IOldMode

Arguments:

IMode: .T. to activate, .F. to deactivate the standby mode.

Return:

A logic value representing the previous state.

Example:

```
// Freeze the Browse: can't move anymore, nor allow other action during the Get edition
oB:StandByMode(.T.)
  lOldReadState := NTK_ReadExit( .T. )
  READ INTO oB:hWnd TOOLTIP
  lGetUpdated := NTK_Updated()
  NTK_ReadExit( lOldReadState )
oB:StandByMode(.F.) // Free the Browse and let it work again...
```

70

0.5.38 **Top()**

0.5.39 $U_p()$

0.6 Properties and Instance variables

0.6.1 alias

Char value. (Access/Assign)

This variable is used to select the correct workarea to use when record/cursor movement or data refresh messages are received. This variable **MUST** be correctly set if you want NtkTBrowse to automatically select the correct database.

Example:

```
oB:Alias := "PRODUCT" // Browsing PRODUCT.DBF
```

0.6.2 autoLite

Logical value. (Access/Assign)

When set to .T., all cells of the current row are highlighted. User can see a complete highlight bar.

When set to .F., only the current cell is that highlights a single cell only.

Default is set to .F.

Example:

```
oB:autoLite := .F. // Highlights current cell only
```

0.6.3 autoSize

Logical value. (Access/Assign)

Determine if yes or no we want to let NtkTbrowse automatically set internal properties based on fonts selected, width of data returned from the column blocks, window size, etc.

The default is .T. - If autoLite is set to .F., the developer have to set column widths, heading height, etc. To manually set column widths and let NtkTbrowse manage the rest, just set saveColSize to .T.

0.6.4 beeper

Logical value. (Access/Assign)

Beep when the user moves the cursor past the top, bottom, left or right most column/row. The default is .T. (On)

0.6.5 boxCursor

Logical value. (Access/Assign)

Mimics Spreadsheet Cursor focus. (Black thin borders) In other words, make the browse cell focus to look like MS Excel or OpenOffice Calc.

0.6.6 cargo

Value of any type. (Access/Assign)

Contains a value of any data type provided as a user-definable slot. NtkTBrowse:cargo allows arbitrary information to be attached to a NtkTBrowse object and retrieved later.

0.6.7 colCount

Numeric value. (Access/Readonly)

A numeric value indicating the amount of columns within the browse.

0.6.8 colFont

Numeric value. (Access/Assign)

A handle to a valid font used to display the data in the browse columns. It is also possible to specify a different font for a specific column by using setColFont().

As developer is responsible for creating or 'getstocking' this font, he is also responsible for its later destruction.

0.6.9 colGridColor

numeric value. (Access/Assign)

Set the color used to draw column grid separators. Should be the result of calling the NTK_RGB() macro. Default is a light gray.

Note: This property has no effect is colGrids is disabled (set to .F.).

0.6.10 colGrids

Logical value. (Access/Assign)

NtkTbrowse can draw Light gray lines (Grid) between each row/col. Default is set to .T. (On)

Note that You can change the default LTGray color using the colGridColor property.

0.6.11 colPos

Numeric value. (Access/Assign)

A numeric value containing the column order where the browse cursor is currently located.
If set directly refreshAll() MUST be sent to update the browse display.

0.6.12 colorSpec

Array value. (Access/Assign)

Unlike Clipper or xHarbour TBrowse, the colorSpec used by NtkTBrowse is an array with the following format:

NTK_RGB() Value for un-selected foreground color / NTK_RGB() value for the un-selected background color

NTK_RGB() value for the selected foreground color / NTK_RGB() value for the selected background color

NTK_RGB() value for the color for the used cells (in any)

Example:

```
oB:colorSpec := { NTK_RGB( 0, 0, 255) , NTK_RGB(255, 255, 176) ,; // UnSel Cells/Item
                  NTK_RGB(255, 255, 255) , NTK_RGB(156, 131, 247) ,; // Sel Cells/Item
                  NTK_RGB(184, 184, 184) ; // WndBrowse fg/bg
                }
```

Note: If you specify a foreground and background color in your column definition then those colors will override the colorspec colors for the un-selected colors for that column.

0.6.13 destroyBlock

Block value. (Access/Assign)

A CodeBlock to be evaluated when the browse is destroyed. Can be used to kill/clean other browses, database closing, and so on...

Example:

```
// Close database on exit
oB:destroyBlock := { |oB| DbCloseArea("PRODUCTS") }
```

0.6.14 doubleclick

Block value. (Access/Assign)

CodeBlock to be evaluated when user doubleclicks or strikes Enter key.

Examples:

```
** Example 1
// Do edit routine
oB:doubleClick := { |oB| EditCell(oB) }

** Example 2
// Do edit routine
oB:doubleClick := { |oB,nMsg,nWparam,nLparam| EditCell(oB) }

** Example 3
// Do edit routine
oB:doubleClick := { |oB,nMsg,nWparam,nLparam| EditCell(oB,nMsg,nWparam,nLparam) }

** Example 4
// A codeblock to be executed when user doubleclicks on current cell/thumbmail
oB:DoubleClick := { || RefreshWhiteArea(hwndMain) }

// ---- For further sample or other ways to use the DoubleClick property, see also:
//      \WNTK4HRB\NTKRAD\BRTEST.PRG
//      \WNTK4HRB\NTKRAD\BRWIMAGE.PRG
```

0.6.15 drawingCol

Numeric value. (Access)

A numeric value representing the column order where the browser is currently drawing or redrawing during its internal painting phases (consequently to invocation of methods like: `Configure()`, `RefreshAll()`, `RefreshCurrent()`, and related...)

0.6.16 **drawingRow**

Numeric value. (Access)

A numeric value representing the line order where the browser is currently drawing or redrawing during its internal painting phases (consequently to invocation of methods like: `Configure()`, `RefreshAll()`, `RefreshCurrent()`, and related...)

0.6.17 escape

Logical value. (Access/Assign)

Let the user the opportunity to exit/terminate (or not) the browse by pressing Esc Key. Default is .T.

0.6.18 headFont

Super vertical Heading Font and horizontal Nubs. Must be a valid handle to a font or a stockObject.

0.6.19 headColor

Super Heading Foreground Color. Should be the result of calling the `NTK_RGB()` macro.

0.6.20 headHeight

This property let you define the height of the 'Nubs' headings.

Please note that due to MS Win32 Listview Api limitations, you can't change the default vertical header size when manifest (for XP styles) is activated.

0.6.21 hitBottom

0.6.22 hitTop

0.6.23 hHeader

Numeric value. (Access/ReadOnly)

Handle to the Browse header control

0.6.24 hNubWnd

Numeric value. (Access/Readonly)

Handle to the Browse left side 'nub' control

0.6.25 hWnd

Numeric value. (Access/ReadOnly)

Handle to the Browse (ListView data) control

0.6.26 keyBlock

Block value. (Access/Assign)

A Code block to be evaluated each time a key is pressed. In other terms when a WM_CHAR, WM_KEYDOWN or WM_SYSKEYDOWN message is received. This block is generally used to call an 'Edit routine' or all other keyboard handle routine...

Examples:

```

** Example 1
// Do edit routine
oB:keyBlock := { |oB| EditCell(oB) }

** Example 2
// Do edit routine
oB:keyBlock := { |oB,nMsg,nWparam,nLparam| EditCell(oB) }

** Example 3
// Do edit routine
oB:keyBlock := { |oB,nMsg,nWparam,nLparam| EditCell(oB,nMsg,nWparam,nLparam) }

** Example 4
oB:KeyBlock := { || NTK_MsgBox( , "LastKey␣pressed␣is␣"+STR(NTK_LastKey()), "info!" )
}

// ---- For further sample or other ways to use the keyBlock property, see also:
//      \WNTK4HRB\NTKRAD\BRWRAD.PRG
//      \WNTK4HRB\NTKRAD\BRTEST.PRG

```

0.6.27 leftVisible

Numeric value. (Access/Assign)

Column number of the left most column currently visible.

Note: This property should not be modified. If you have to do so, take care of unexpected results.

0.6.28 lineHeight

Numeric value. (Access/Assign)

Set manually the height of the browse rows.

100

0.6.29 make3D

Logical value. (Access/Assign)

3D look : Each cell appears to be raised. Defaults is set to .F.

Example:

```
oB:make3D := .T. // Make a fancy 3D browse...
```

0.6.30 `mimicButton`

Logical value. (Access/Assign)

Vertical and Heading nubs will work like buttons. Defaults is set to .F.

Example:

```
oB:mimicButton := .T. // nubs will work like buttons
```

0.6.31 msgBlock

Block value. (Access/Assign)

Not Implented yet.

A Code block to be evaluated when the browse window receives a message.

For example this property can be use to send synchronisation messages between several browse objects. Of course, there is certainly many other situations in which to use it...

0.6.32 nCrecNo

Numeric value. (Access/Assign)

Current physical record number of the database being browsed: RecNo()

Developer may also change this value when needed, but it is not recommended as NtkTBrowse is able to control it automatically.

0.6.33 nCurRec

Numeric value. (Access/Assign)

Current Logical record position for thumb position calculation. This property is also used to keep track of the current record position within `NtkTBrowse()`. The developer is responsible for setting this instance variable.

See also **nMaxRec** instance variable.

0.6.34 nMaxRec

Numeric value. (Access/Assign)

Current number of records in the database being browsed: LastRec()

The developer is responsible for updating this value each time a record is added or deleted.

See also **nCurRec** instance variable.

Example :

```
(oB:alias)->( DbDelete() )           // Delete current record
oB:nMaxRec := (oB:alias)->( LastRec() ) - 1 // Update record count
oB:nCrecNo := (oB:alias)->( RecNo() )       // Update current physical record n
oB:nCurRec := oB:nCurRec - 1             // Update current logical record (t
oB:refreshAll()                        // refresh browse object
```

0.6.35 nubAlign

// Nub text or bitmap alignment. ([DT_CENTER], DT_LEFT, DT_RIGHT)

0.6.36 nubBlock

Block value. (Access/Assign)

A codeblock. Evaluated for each record read.

Note: This property has no effect if ShowNumber is disable (.F.).

Example:

```
oB:nubBlock := {|| IIF( CLIENT->NATURE == 'L', "Local", "Export" ) }
```

0.6.37 nubHeadBlock

Block value. (Access/Assign)

A codeblock to be evaluated when the user clicks on each "nub" column header.

Note: This property has no effect if mimicButton is disable (.F.).

Example:

```
mimicButton := .T. // Column & header nubs will mimic std button behaviour.  
nubHeadBlock := {|ob| SetHeaderTick(ob)} // Display a bitmap the clicked header.
```

0.6.38 nubHheadBlock

Block value. (Access/Assign)

A codeblock to be evaluated when the user clicks on the right most "nub" Heading

0.6.39 nubHBitmap

// Display a bitmap on the nub heading. handle to a bitmap in your res file or one of the standard Windows bitmaps.

0.6.40 nubHBitAlign

// Nub bitmap heading alignment. ([DT_CENTER], DT_LEFT, DT_RIGHT)

0.6.41 nubClickBlock

Block value. (Access/Assign)

A codeblock to be evaluated if the user clicks on a "nub"

0.6.42 nubColor

Numeric value. (Access/Assign)

NTK_RGB() which is used to set the textcolor for the "nubs".

0.6.43 nubHeading

Char or Array value. (Access/Assign)

A character constant/expression/array to be displayed on the "nub" column heading

0.6.44 nubHeadClickPos

Numeric value. (Access/Readonly)

Value indicating the last nub's head clicked column.

0.6.45 nubHeadColor

Numeric value. (Access/Assign)

NTK_RGB() which is used to set the textcolor for the "nub" heading.

0.6.46 nubWidth

Numeric value. (Access/assign)

Used to manually set the width of the "nubs". Value is in pixels.

Due to Listview Win32 API limitations, this property can be set only once: At first initialization.

If you try to modify its value afterward, then calling the Configure() method, your changes will not affect the current running Browse.

0.6.47 onGainFocus

Block value. (Access/Assign)

A code block to be evaluated each time the browser gains the focus.

0.6.48 onLooseFocus

Block value. (Access/Assign)

A code block to be evaluated each time the browser loses the focus.

120

0.6.49 rightClick

Block value. (Access/Assign)

A code block to be evaluated each time the user right clicks the mouse

0.6.50 rightVisible

Numeric value. (Access/Assign)

Column number of the right most column currently visible.

Note: This property should not be modified. If you have to do so, take care of unexpected results.

122

0.6.51 rowCount

Numeric value. (Access/ReadOnly)

How many visible rows in browse?

0.6.52 rowGridColor

numeric value. (Access/Assign)

Set the color used to draw line grid separators. Should be the result of calling the NTK_RGB() macro. Default is a light gray.

Note: This property has no effect is rowGrids is disabled (set to .F.).

0.6.53 rowGrids

Logical value. (Access/Assign)

NtkTbrowse can draw light gray lines (Grid) between each row/col. Default is set to .T. (On)

Note that You can change the default LTGray color using the RowGridColor property.

0.6.54 rowPos

Numeric value. (Access/Assign)

A numeric value containing the row order where the browse cursor is currently located. If set directly refreshAll() MUST be sent to update the browse display.

0.6.55 saveColSize

Logical value. (Access/Assign)

Just for syntax backward compatibility with C4W.

0.6.56 sbHorz

Logical value. (Access/Assign)

Force to display or hide the Horizontal ScrollBar working with the browse control.

Note: This property supersede any WS_HSCROLL that could be previously added to the Style parameter of NtkTbrowse():Init() method.

0.6.57 sbVert

Logical value. (Access/Assign)

Force to display or hide the vertical ScrollBar working with the browse control.

Note: This property supersede any WS_VSCROLL that could be previously added to the Style parameter of NtkTbrowse():Init() method.

0.6.58 selectblock

Block value. (Access/Assign)

A code block to be evaluated each time the browse cursor is re-positioned.

The browse object, the current logical record number (nCurRec), and .T. or .F.

(depending on whether or not the browse window has focus) is automatically passed to the code block.

Example:

```
// Call the Update/Change procedure...  
oB:selectBlock := { |oB,nRecNo,lFocus | OnMove(oB,nRecNo,lFocus) }
```

0.6.59 showFocus

Logical value. (Access/Assign)

If set to .T. NtkTBrowse will draw a focus rectangle around the currently selected cell.
Default is set to .F.

Note that this property has no effect when make3D is active (.T.)

0.6.60 showNumbers

Logical value. (Access/Assign)

Show logical record number on Nubs. Default is set to .T.

132

0.6.61 singleClick

Block value. (Access/Assign)

A codeBlock to be evaluated when user single click.

0.6.62 **tbFont**

Numeric value. (Access/Assign)

Default Font used to display all data cells of the browse. Must be a handle to a valid Windows font or a stockobject. As developer is responsible for creating or 'getstocking' this font, he is also responsible for its later destruction.

0.6.63 userMove

Logical value. (Access/Assign)

Allow user to move a column by clicking and dragging it to its new position. Default is set to .T. (On).

0.6.64 userSize

Logical value. (Access/Assign)

Allow user to resize a column by clicking the space between the column headings and then "dragging" the line to the left or right. Default is set to .T. (On).

0.7 Related functions.

0.7.1 NtkTBGetCurrent(void)

Gives the handle to the current active browse (if any). Very Useful in a multiple browse environment.

Syntax:

NtkTBGetCurrent() -> hWndCurTB

Arguments:

None.

Return:

A numeric value representing the handle to the current active browse. If there's no active browse, zero is returned.

Example:

```

hWndCurTB := NtkTBGetCurrent()

IF oB1:hWnd == hWndCurTB
  oB1:make3D := IIF( oB1:make3D, .F., .T. )
  oB1:Configure() // Reconfigure oB1 again!
  NTK_SetFocus(oB1:hWnd)
ELSEIF oB2:hWnd == hWndCurTB
  oB2:make3D := IIF( oB2:make3D, .F., .T. )
  oB2:Lineheight := IIF( oB2:make3D, 24, 18 )
  oB2:Configure() // Reconfigure oB2 again!
  NTK_SetFocus(oB2:hWnd)
ENDIF

```

Part IV
Miscellaneous.

0.8 Picture and Bitmap support.

NtkTBrowse supports the use of bitmaps within column headings, nubs (left side headers), and in browse columns (data cells).

Those Bitmaps **MUST** be included in your application's resource file and must use numeric IDs.

Though, in browse columns, instead of a numeric ID (referring to the resource bitmap), you can pass a string specifying the complete pathfolder and name of the picture to be displayed. (i.e "C:\MyAlbum\mypict.jpg")

Please note that it is also possible to call internal Windows bitmaps (OBM_CHECK, OBM_CLOSE, OBM_*, etc...).

You can specify bitmaps for column headings in the column object or by calling setColBitMap().

If you're using bitmaps into the nubs or column headings, the nubBlock **MUST** return the numeric ID of bitmap being used.

Notes:

a) If you want the background color of the bitmap to be treated as transparent, just fill it (i.e with GIMP) using the magenta colour (RGB: 255,0,255). NtkTbrowse directly understands that!

b) Bitmaps and text can be both mixed in data cells, using the Block and Bitmap instance vars. For further details on how to do, give a try to:

```
\WNTK4HRB\NTKRAD\BRTEST.prg
```

```
\WNTK4HRB\NTKRAD\BRWIMAGE.prg
```

0.9 Default navigation keys and mouse behaviour.

The following keys are used internally by NtkTBrowse to move the browse cursor and record pointer position:

```

Alt-End      - Moves the cursor to right most column.
Alt-Home     - Moves the cursor to the left most column.
Ctrl-End     - Moves the cursor to first column. (pan End)
Ctrl-Home    - Moves the cursor to the last column. (pan Home)
Up Arrow     - Moves the record pointer up one record.
Down Arrow   - Moves the record pointer down one record.
End          - Moves the record pointer to the last record.
Home        - Moves the record pointer to the first record.
Left Arrow   - Moves the cursor left one column.
Right Arrow  - Moves the cursor right one column.
Ctrl-LFArrow - Moves the cursor left one page. (pan Left)
Ctrl-RTArrow - Moves the cursor right one page. (pan Right)
PgUp        - Moves the record pointer up one page.
PgDn        - Moves the record pointer down one page.
Enter       - Evaluates the doubleClick code block if not NIL.

```

The following mouse behavior are used internally by NtkTBrowse to move the browse cursor position:

```

Left Mouse Double click - Evaluates the doubleClick code block if not NIL.
Left Mouse Single click - Evaluates the singleClick code block if not NIL.
MouseWheel Up          - Moves the record pointer up one record.
MouseWheel Dn          - Moves the record pointer up one record.
Right Mouse Single click - Evaluates the rightClick code block if not NIL.

```

0.10 NtkTBrowse restrictions.

As you can notice reading this document, NtkTBrowse is a powerful, highly configurable and really an easy to use object class.

Frankly speaking , NtkTBrowse has been tested using large DBF of several hundred thousand records, with pretty good loading and drawing access times, regarding to the lowest and oldest machine we still have at our office (i.e Athlon 1 GHz, RAM 512MB, HDD 40GB, Video nVidia 64MB) .

We only encountered slight slackening in horizontal moves, when using databases with huge amount of (small size) fields. But anyway, never it has been something painful.

However, if you experiment low speed or performance browse issues:

Try to reduce the number of visible rows and/or columns (increasing the `lineHeight` property and/or by defining a smaller browse area). Consequently, this will force NtkTBrowse to adapt its internal caches. Remember, less you have lines and cols to redraw, more faster will be the Listview!

Index

AddColumn() , 34
Align , 16
BgColor , 15
bitalign , 19
Bitmap , 18
Block , 12
Bottom() , 35
Cargo , 21
Configure() , 36
DelColumn() , 37
Destroy() , 39
Down() , 40
End() , 41
fgColor , 14
FindColumn() , 38
Font , 22
GetCellData() , 42
GetCellRect() , 43
GetColCargo() , 44
GetColumn() , 45
GetNubData() , 46
GoBottom() , 47
GoTop() , 48
HeadAlign , 25
HeadBlock , 20
HeadBmpAlign , 27
HeadBmpIndex , 26
HeadBmpVAlign , 28
HeadColor , 23
HeadFont , 24
Heading , 11
Home() , 49
Init() , 31
InsColumn() , 50
Left() , 51
MakeColVisible() , 52
ModColumn() , 53
MoveColumn() , 54
NtkColumn:Init() , 7
NtkTBGetCurrent() , 136
PageDown() , 55
PageUp() , 56
PanEnd() , 57
PanHome() , 58–60
Picture , 13
RefreshAll() , 61
RefreshCurrent() , 62
Right() , 63
SetColWidth() , 64
SetRowHeight() , 65
ShowAll() , 66
ShowCurrent() , 67
ShowData() , 68
StandByMode() , 69
Top() , 70
Up() , 71
Width , 17

alias, 72
autolite, 73
autoSize, 74

beeper, 75
boxCursor, 76

- cargo, 77
- colCount, 78
- colFont, 79
- colGridColor, 80, 123
- colGrids, 81
- colorSpec, 83
- colPos, 82

- destroyBlock, 84
- doubleclick, 85
- drawingCol, 86
- drawingRow, 87

- escape, 88

- headColor, 90
- headFont, 89
- headHeight, 91
- hitBottom, 92
- hitTop, 93
- hWnd, 94–96

- Intro, 3

- keyBlock, 97

- leftVisible, 98
- lineHeight, 99

- make3D, 100
- mimicButton, 101
- msgBlock, 102

- nCrecNo, 103
- nCurRec, 104
- nMaxRec, 105
- nubAlign, 106
- nubBlock, 107
- nubClickBlock, 112
- nubColor, 113
- nubHBitAlign, 111
- nubHBitmap, 110
- nubHeadBlock, 108
- nubHeadClickPos, 115
- nubHeadColor, 116
- nubHeading, 114
- nubHheadBlock, 109
- nubWidth, 117

- OnGainFocus, 118
- OnLooseFocus, 119
- Overview, 3

- rightClick, 120
- rightVisible, 121
- rowCount, 122
- rowGrids, 124
- rowPos, 125

- saveColSize, 126
- sbHorz, 127
- sbVert, 128
- selectblock, 129
- showFocus, 130
- showNumbers, 131
- singleClick, 132

- tbFont, 133

- userMove, 134
- userSize, 135