

TMS Advanced Smooth ListBox DEVELOPERS GUIDE

December 2008
Copyright © 1998 - 2008 by tmssoftware.com bvba
Web: <http://www.tmssoftware.com>
Email: info@tmssoftware.com

Index

TMS Advanced Smooth ListBox availability.....	3
TMS Advanced Smooth ListBox use	3
TMS Advanced Smooth ListBox organisation.....	3
The visual organisation of TMS Advanced Smooth ListBox.....	3
The programmatic organisation of TMS Advanced Smooth ListBox.....	5
TMS Advanced Smooth ListBox important methods and properties.....	7
Categories.....	7
Fill property	7
Office style property.....	9
ShowDetails and HideDetails methods	9
Y-Position to Item	10
Item Selection.....	10
Drag & drop items within the listbox	10
Smooth progressbar and button within the listbox	10
Item Grouping	11
TMS Advanced Smooth ListBox important events.....	13
TMS Advanced Smooth ListBox keyboard and mouse support	14

TMS Advanced Smooth ListBox availability

TMS Advanced Smooth ListBox is available as VCL component for Win32 application development.

VCL versions:

TMS Advanced Smooth ListBox is available for CodeGear™ Delphi 6, 7, 2005, 2006, 2007, 2009 & CodeGear™ C++Builder 2006, 2007, 2009

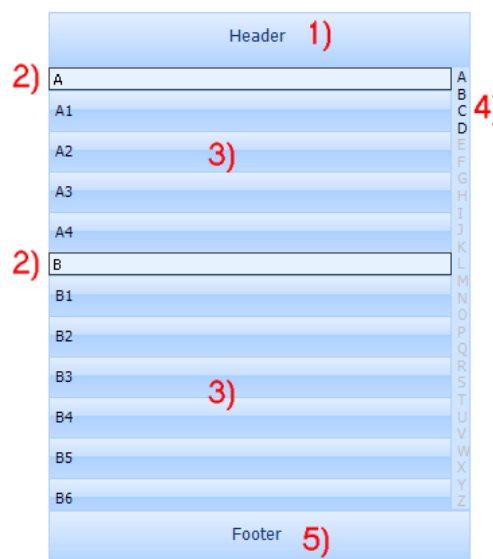
TMS Advanced Smooth ListBox use

TMS Advanced Smooth ListBox is designed to easily navigate through fully customizable list items with many built-in features from item grouping and sections to keyboard lookup and animated mouse scrolling.

TMS Advanced Smooth ListBox organisation

The visual organisation of TMS Advanced Smooth ListBox

TMS Advanced Smooth ListBox is a component with the following main visual elements:



- 1) The header can contain information on how the items are organised or which types of items are in the ListBox. For example: "Contact list". The header can also contain HTML text and images. The header can be visible or not.
- 2) In case you define the ListBox as a "Contact list" you can set the item sections visible. This allows you to group the items under the section of the same category. The categories which are currently available are alphabetical and numeric.
- 3) These are the ListBox items which can be fully customized in many ways: HTML text, images graphics, controls ...
- 4) When you want to search an item you can easily navigate to the first item with the letter you clicked at the lookup bar.
- 5) The optional footer is identical to the header and allows displaying extra information at the bottom of the list, such as summary information for example like "Number of contacts: ..."

The ListBox item can be customized in many ways: various text elements, graphics, and controls. Below is an overview of the important visual elements of the ListBox item.



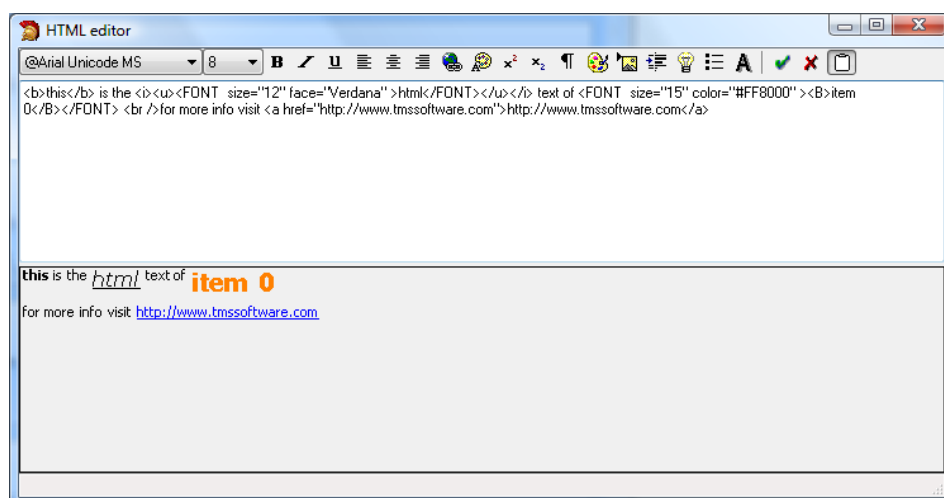
- 1) The item can contain a graphic on the left and the right side. The type of graphic can be changed with the property GraphicLeftType or GraphicRightType.

The types you can choose from are listed below. In case of gtDetailImage the image can be clicked to enter the details. When choosing gtCommonImage or gtCommonDetailImage, the function is the same as the gtImage or gtDetailImage type, but only uses a global image defined on TAdvSmoothListBox level that can be used for all items.

Supported types:

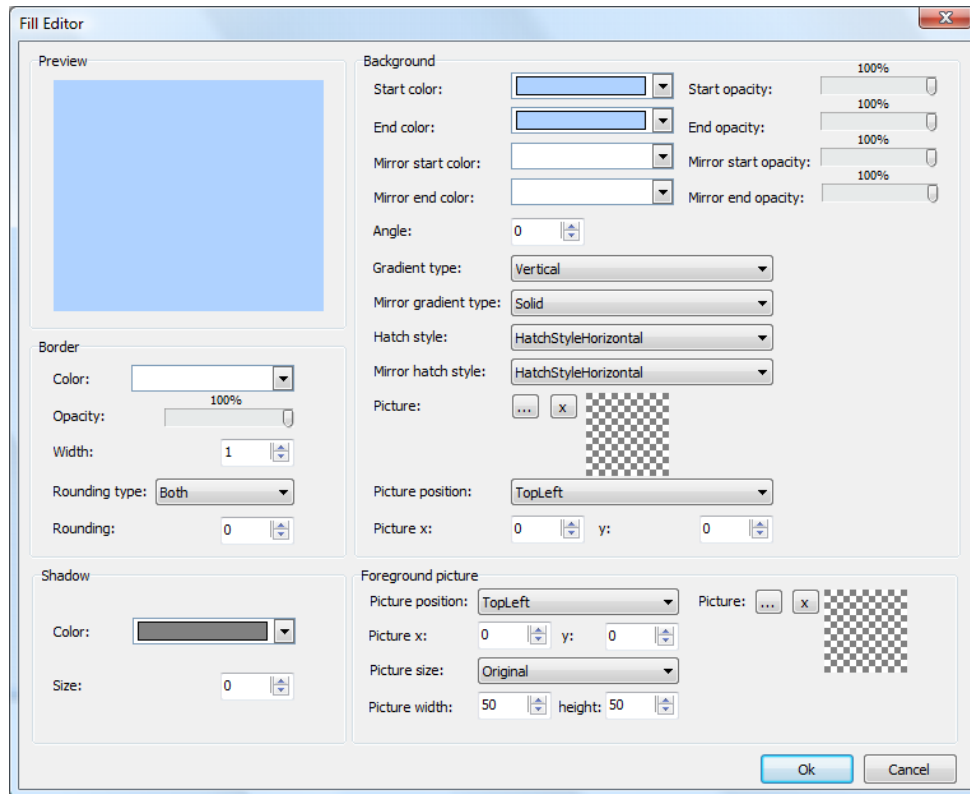
- gtButton, gtSmoothButton: button is displayed on the item
- gtCheckBox: checkbox is displayed on the item
- gtRadioButton: radiobutton is displayed on the item
- gtImage: image is displayed on the item
- gtDetailImage: image that shows detail control on click is displayed
- gtCommonImage: common listbox image is displayed on the item
- gtCommonDetailImage: common listbox detail image is displayed on the item
- gtNone: nothing is displayed

- 2) The caption text of the item. The caption can optionally be clicked and triggers the OnItemCaptionClick event.
- 3) The HTML notes text of the item can be formatted with a design time HTML editor. The HTML can contain images listed in a TGDIPictureContainer or a TImageList. When clicking on an anchor of the HTML text the event OnAnchorClick is called.



- 4) The info of the item can contain extra information of the info and when clicked will trigger the OnItemInfoClick event.

Most of the elements are drawn with the GDIPfill class (included in the source of the component). Every time you want to edit the style of the header or the lookupbar you will get a design time editor with a preview which allows you to fully customize the look of the component.



The details of the fill are explained later in a separate paragraph that discusses this.

These are the most important elements of the ListBox component:

The programmatic organisation of TMS Advanced Smooth ListBox

In design time the AdvSmoothListBox has some items by default. Below you can see how to add items in code and in some cases extra properties can be set to change the item appearance.

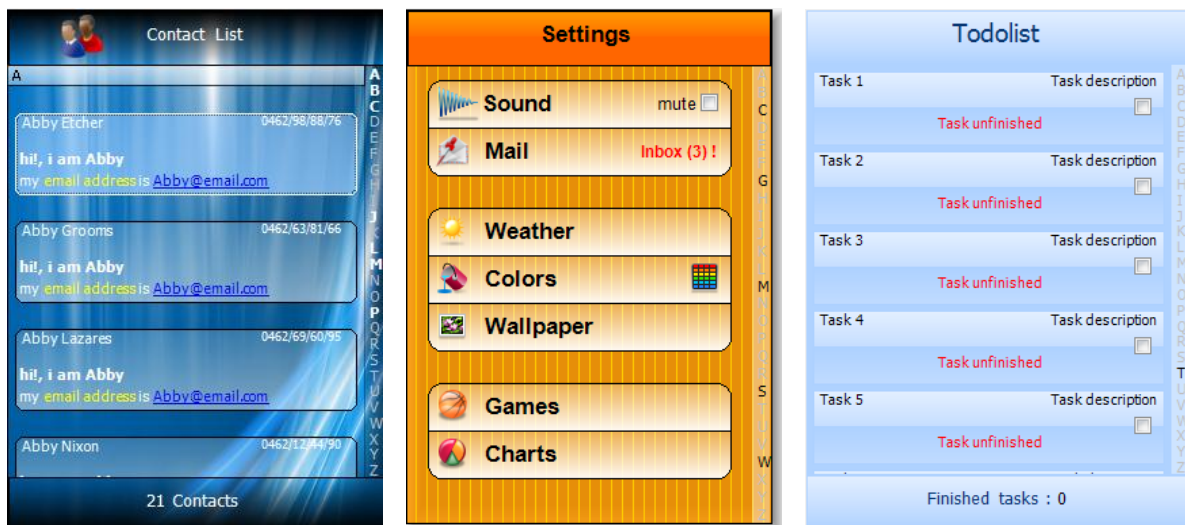
```
procedure TForm1.AddPoints;
var
  i: integer;
begin
  //Adds 20 items
  for i := 0 to 20 do
  begin
    with AdvSmoothListBox.Items.Add do
    begin
      Enabled := True; //If enabled false then the item will be drawn in
      disabled fill
      GraphicLeftType := gtCheckBox
      Checked := False; //the item will be checked if the type is a
      checkbox or a radiobutton
      Caption := 'Item ' + inttostr(i);
```

```

Info := 'Info ' + inttostr(i);
Notes := 'Notes ' + <a href="http://www.tmssoftware.com">link</a>
Splitter := false; //if splitter is true a empty space will be drawn
between 2 items
DetailControl := Panel1 //depending on the action chosen to visualize
the details the detailcontrol will be shown.
Level := 1; //Set the level property to allow expanding and
collapsing item groups
Indent := 30; //Set the indent property to visually indent the item.
end;
end;
end;

```

Some screenshots of the AdvSmoothListBox after applying different styles.



TMS Advanced Smooth ListBox important methods and properties.

Categories

On the right or left side of the control, a lookup bar can be shown with categories to navigate to the first item of that specific clicked category. You can choose between two types of categories: the built-in alphanumeric categories or custom categories. When you choose the custom category type you can add custom category items to the TAdvSmoothListBox.Categories collection. Each category has a category ID, text and imageindex to optionally show an image from the TAdvSmoothListBox imagelist. An item of the TAdvSmoothListBox can be simply assigned to a category by setting the item's CategoryID property. Below is a sample after you apply custom categories.



Further customization of the category display on the lookup bar can be found under TAdvSmoothListBox.LookupBar.

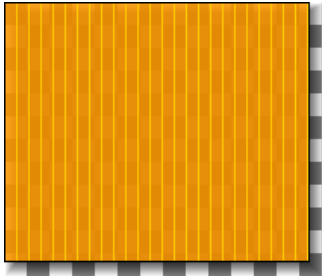
Fill property

Every element which implements the TGDIPFill class can use these properties to change the look.

- **Color:** the start color of the gradient (in case gtSolid Color is the only property used).
- **ColorTo:** the end color of the gradient.
- **ColorMirror:** when ColorMirror is set to a color different from clNone the fill will be split up in 2 parts: the normal upper part and the mirror bottom part. ColorMirror is the start color of the mirror part.
- **ColorMirrorTo:** the end color of the mirror part.
- **GradientType:** the type of gradient to apply: Horizontal, Vertical, Angle, Hatch...
- **GradientMirrorType:** the type of gradient of the mirror part.
- **HatchStyle:** the hatchstyle of the gradienttype in case gtHatch is chosen.

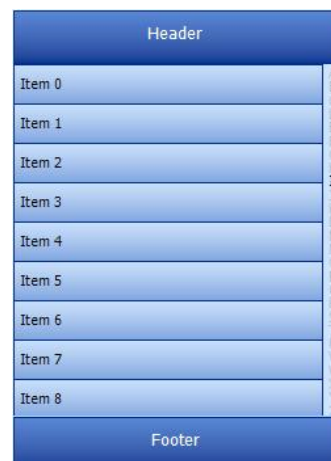
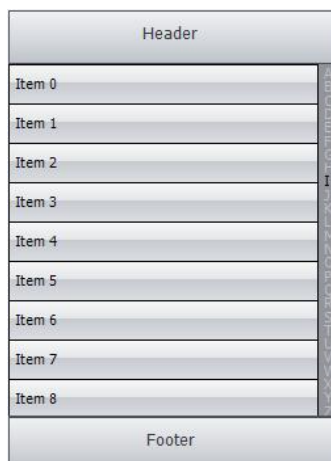
- **HatchStyleMirror**: the hatchstyle of the mirror part.
- **BackgroundPicture**: the backgroundpicture in case gtTexture is chosen. The backgroundpicture can be stretched or positioned on the fill.
- **BackgroundPicturePosition**: the position of the backgroundpicture.
- **BackgroundPictureLeft**: when the position is set to custom the left position can be set with this property
- **BackgroundPictureTop**: the top position of the backgroundpicture when the position is set to custom.
- **Picture**: you can always set a picture that is not bound to the rectangle of the fill. In other words you can draw a picture which overlaps the fill.
- **PicturePosition**: the position of the picture.
- **PictureLeft**: the left position of the picture in case pictureposition is set to custom.
- **PictureTop**: the top position of the picture in case pictureposition is set to custom.
- **PictureSize**: it can be useful to resize the picture to a different size when it is too large. Set picturesize to custom and use picturewidth and pictureheight to change the size of the picture.
- **PictureWidth**: the width of the picture in case the picturesize is set to custom.
- **PictureHeight**: the height of the picture in case the picturesize is set to custom.
- **Opacity**: the opacity of the start color of the normal part.
- **OpacityTo**: the opacity of the end color of the normal part.
- **OpacityMirror**: the opacity of the start color of the mirror part.
- **OpacityMirrorTo**: the opacity of the end color of the mirror part.
- **BorderColor**: the color of the border of the fill.
- **BorderOpacity**: the opacity of the border of the fill.
- **BorderWidth**: the width of the border of the fill.
- **Rounding**: the rounding of the fill, set Rounding = 0 to have a rectangular shape.
- **RoundingType**: the type of rounding of the fill. In some cases it can be useful to only set the top corners of the fill to be rounded.
- **ShadowColor**: the color of the shadow of the fill.
- **ShadowOffset**: the offset of the shadow of the fill.
- **Angle**: the angle of the gradient in case gtAngle is used.

Some examples of fill styles:



Office style property

TAdvSmoothListBox has built-in office 2003 and office 2007 styles. Just right-click on the component at design-time and choose Styles.



At runtime, the component is TAdvFormStyler-aware. When using a TAdvFormStyler and changing its style, the TAdvSmoothListbox will change style along all other controls on the form.

ShowDetails and HideDetails methods

When clicking on an item with a graphic DetailImage, the detail control will show. But you can also call ShowDetails in code. Just call AdvSmoothListBox.ShowDetails and the ListBox will show the detail control associated with the selected item. For hiding the details call AdvSmoothListBox.HideDetails. You can check the detailstatus of the ListBox by calling AdvSmoothListBox.DetailStatus.

You can call the details on many different ways, either with the mouse or with the keyboard (See: Advanced Smooth ListBox Keyboard support).

There are 2 properties that can be set to change the way the details are shown. For the mouse set ShowDetailClick to the type you want.

For the keyboard use the property ShowDetailKey.

Y-Position to Item

You can find an item by using `YToltem` or `ItemAtXY`. The function returns the index of the item that was found at the location of the Y-position. The function returns -1 when no item was found.

Get the first and the last visible items.

Call `GetTopIndex` or `GetBottomIndex` to get the item displayed at the top of the listbox or at the bottom of the listbox.

Item Selection

By default, `TAdvSmoothListBox` works with single selection. The selected item is set via `TAdvSmoothListBox.SelectedItemIndex`. When `SelectionMode` is set to `sAutoDeselect`, selection disappears upon scrolling (like the iPhone list). When `SelectionMode` is set to `sPersistSelection`, selection is persisted while scrolling. A variant on the `sPersistSelection` mode is the `sPersistSelectionAlways` mode, with this mode you can multiselect items without the keyboard.

When the property `MultiSelect` is set true, multiple items can be selected either with the mouse or the keyboard. Just Shift click on an item if you want to select each item between the previous and the current selected item, or ctrl click on an item if you only want the clicked item to be added to the selection. To get or set which items are selected, the property `Item.Selected` can be used.

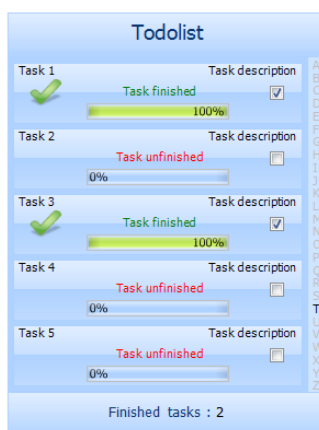
Drag & drop items within the listbox

When property `ItemDragging` is set to true, you can drag drop items to a different position in the list by Alt - click (and holding Alt key) on the item. The item will be draggable and can be inserted on the position where the item is above.

Smooth progressbar and button within the listbox

A progressbar and a button can be added for to add extra functionality on the listboxitem. Simply select `gtSmoothButton` as `GraphicLeft` or `GraphicRight` to add a smooth button. To add a progressbar, set the property `ProgressBarVisible` to true. The progressbar minimum, maximum and position can also be set on each item. The progressbar and button appearance can be set in the `itemappearance` property on listbox level.

Sample



Item Grouping

With the AdvSmoothListBox it is possible to group the items and expand or collapse them with a single click. The level property on an item allows you to define the different groups to collapse or expand. The indent property can be used to visually create a “node” that is indented to the right. An item has a public expanded property. With this property you can expand or collapse items with a level that is higher than the current item.

Sample

Three items with level 0 Car brand and for each card brand there are 3 items with a level 1 and an indent of 30.

Code

```
with AdvSmoothListBox5.Items.Add do
begin
  Caption := 'Mercedes';
  GraphicLeftType := gtSmoothButton;
  ButtonCaption := '-';
  ButtonBevelColor := clBlack;
  ButtonColor := clGreen;
  GraphicLeftWidth := 25;
end;
with AdvSmoothListBox5.Items.Add do
begin
  Level := 1;
  Caption := 'Mercedes SLK Roadster';
  Indent := 30;
end;
with AdvSmoothListBox5.Items.Add do
begin
  Level := 1;
  Caption := 'Mercedes SLR Coupé';
  Indent := 30;
end;
with AdvSmoothListBox5.Items.Add do
begin
  Level := 1;
  Caption := 'Mercedes GLK 4x4';
  Indent := 30;
end;

with AdvSmoothListBox5.Items.Add do
begin
  Caption := 'BMW';
  GraphicLeftType := gtSmoothButton;
  ButtonCaption := '-';
  ButtonBevelColor := clBlack;
  ButtonColor := clGreen;
  GraphicLeftWidth := 25;
end;
with AdvSmoothListBox5.Items.Add do
begin
  Level := 1;
  Caption := 'BMW M3';
  Indent := 30;
end;
with AdvSmoothListBox5.Items.Add do
```

```
begin
    Level := 1;
    Caption := 'BMW Z4';
    Indent := 30;
end;
with AdvSmoothListBox5.Items.Add do
begin
    Level := 1;
    Caption := 'BMW X5';
    Indent := 30;
end;

with AdvSmoothListBox5.Items.Add do
begin
    Caption := 'Land Rover';
    GraphicLeftType := gtSmoothButton;
    ButtonCaption := '-';
    ButtonBevelColor := clBlack;
    ButtonColor := clGreen;
    GraphicLeftWidth := 25;
end;
with AdvSmoothListBox5.Items.Add do
begin
    Level := 1;
    Caption := 'Land Rover Defender 90';
    Indent := 30;
end;
with AdvSmoothListBox5.Items.Add do
begin
    Level := 1;
    Caption := 'Land Rover Series III';
    Indent := 30;
end;
with AdvSmoothListBox5.Items.Add do
begin
    Level := 1;
    Caption := 'Range Rover Sport V8';
    Indent := 30;
end;
```

Result



TMS Advanced Smooth ListBox important events

OnAnchorClick: when HTML hyperlinks are used in the header or footer and this hyperlink is clicked, the event OnAnchorClick is triggered and returns the anchor value.

OnHideDetail: event triggered when hiding the detail control of an item.

OnItemAnchorClick: when HTML hyperlinks are used in the listbox item and this hyperlink is clicked, the event OnItemAnchorClick is triggered and returns the item and anchor value.

OnItemBkgDraw: event triggered when drawing the background of the item. Set parameter DefaultDraw = true if you want that default drawing is also done by the TAdvSmoothListBox itself.

OnItemCaptionClick: event triggered when clicking on the caption of an item.

OnItemButtonClick: event triggered when clicking on the button of an item.

OnItemCheckClick: when adding checkboxes to the items this event will be triggered when clicking on a checkbox.

OnItemClick: event triggered when clicking on the item.

OnItemDbClick: event triggered when double-clicking on the item.

OnItemDraw: event triggered when drawing the item. This event can be used to customize the item appearance. Set parameter DefaultDraw = true if you want that default drawing is also done by the TAdvSmoothListBox itself.

OnItemImageClick: when adding an image as a left or right graphic in the item, this event will be triggered if you click on the image.

OnItemInfoClick: event triggered when clicking on the info.

OnItemRadioClick: when adding radio buttons this event will be triggered if you click on the radio button.

OnItemText: event called when drawing text on the item. The text can be changed dynamically through this event.

OnShowDetail: event called when showing the detail control of an item.

OnItemMouseLeave / OnItemMouseEnter: event called when entering or leaving an item.

OnItemHint: event called when hovering an item and the showHint property is true. A hint can be set on every item with the item.Hint property.

OnHeaderClick / OnFooterClick: Event called when clicking on the footer or the header.

OnLookupClick: Event called when clicking on the lookupbar.

OnSelectionChanged: Event called when item selection changes.

OnItemDragOver: Event called when an item is dragged over another item.

OnItemDragEnd: Event called when dropping an item outside of the area of the items.

OnItemDragDrop: Event called when dropping an item inside of the area of the items.

OnItemDragStart: Event called when alt-clicking on an item and moving the mouse up or down.

TMS Advanced Smooth ListBox keyboard and mouse support

Keyboard

When tabstop is set to true, full keyboard support is enabled for listbox. Below is a list of keys that will allow you to navigate through the listbox without using the mouse.

First of all you can show and hide the detail control of an item by pushing the key set in the ShowDetailKey property of the TAdvSmoothListBox.

Keys which can be used to show or hide the details:

- Space
- F2
- Return
- None (no detailkey)

The keys below are used to navigate through the AdvSmoothListBox.

- Arrow key down : Scroll one item down
- Arrow key up: Scroll one item up
- Home key: Scroll to the first item
- End key: Scroll to the last item
- Page down key: scroll down in steps of 5 items
- Page up key scroll up in steps of 5 items
- Ctrl - space : toggle selection of the item in multi selection mode

Mouse

A drag-release mouse move allows you to scroll up or down through the items. Like the keyboard you can use the mouse to enter the details of the item. With the ShowDetailClick property you can set the way you have to click before the detail of the item is showing.

These are the values you can set:

- sdOnClick: detail control is shown upon click in the entire item rectangle
- sdOnDbClick: detail control is shown upon double click in the entire item rectangle
- sdOnDetailImageClick: detail control is shown upon click on the detail image (left or right)
- sdOnDetailImageDbClick: detail control is shown upon double click on the detail image (left or right)