TMS TAdvSearchList - TAdvSearchEdit DEVELOPERS GUIDE

July 2019

Web: https://www.tmssoftware.com
Email: info@tmssoftware.com

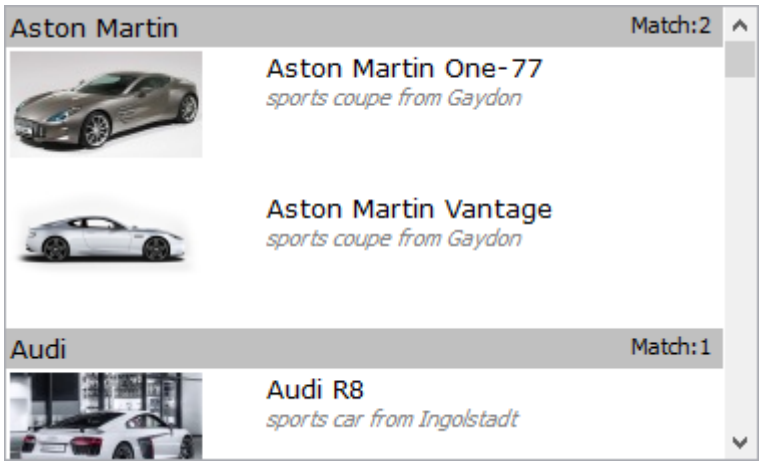
TAdvSearchList & TAdvSearchEdit

Introduction

TAdvSearchList is a list control that has built-in support to display information in multiple columns, it can display images along with text and an item caption and description text and it can show custom drawn items as well. Along with items, it has optional support for categories and items can be displayed grouped into categories. The main feature is that a search can be performed on the list, the matching text from a search can be highlighted and items are filtered automatically that match the search criteria. TAdvSearchEdit is basically an edit control with a dropdown TAdvSearchList and while typing in the edit control, a search/filter is performed in the dropdown TAdvSearchList.

Architecture

Data for the TAdvSearchList or TAdvSearchEdit is organized via the Items collection. An item can be a category header or a regular item. An item has a list of data objects per column. Other than this, there is a Columns collection that holds the number of columns in the list and the properties of each column. Finally, there is a categories property that optionally holds a list of categories. Each item can be assigned to a category and filtering is possible not only on text but also on category.



In this screenshot, we have a TAdvSearchList with two columns. We see 5 items: 2 items of the category type and shown in gray as header for regular items. The regular items have data for 2 columns, one column is a column holding a picture, the other a caption and a description.

The Columns, Items and Categories collections in detail

Columns

AdvSearchList.Columns is a collection of TColumnItem objects and the TColumnItem object has following properties:

Color: TColor
Sets the background color of the column. When the color is clNone, the control's background color is used.
ControlFont: Boolean
When true, the TAdvSearchList.Font is used for the control as opposed to Columns[].Font

Font: TFont
Sets the font to use for the column. Only when Column[].ControlFont = false, this column font is used, otherwise the global control font is used.

Tag: NativeInt
Generic tag property

Visible: Boolean
When true the column is visible in the TAdvSearchList

Width: integer
Sets the width for the column

Items

Items is the collection that holds all items visible in the TAdvSearchList, TAdvSearchEdit dropdown. It is a collection of TSearchListItem objects with following properties:

CategoryID: integer
When a category is used, the numeric ID of the category is set via this CategoryID property. This refers to the ID of the category as added in the control's Categories collection.

Columns: TSearchColumnItems
Collection of TSearchColumnItem objects holding the data for the various columns in the list

The TSearchColumnItem has following properties:

Caption: string
Sets the caption text for an item's column.

Color: TColor
Sets the color of the item's column. Default this is clNone and uses the column's global background color. When different from clNone, just this column's item will use this background color.

Description: string
Sets the description text, displayed under the caption for an item's column.

ImageIndex: Integer;
Sets the index of the image in an associated imagelist if the column is supposed to display an imagelist image.

Picture: TPicture
Sets the picture to be displayed in the item's column.

Shape: TColumnItemShape
Sets the shape to draw for an item's column. It specifies whether the entire item's column backround is drawn in the background color or just a rectangle or left-side bar.

TextColor: TCcolor
Sets the text color of the item's column. Default this is clNone and uses the column's global font color color. When different from clNone, just this column's item will use this as font color.

ItemType: TSearchItemType
Is by default itItem or a regular search list item or set to itCategory to be displayed as a category header.

Tag: NativeInt
Generic tag property

Note: to load a high number of items at once in the TAdvSearchList or TAdvSearchEdit, it is recommended to use

```
AdvSearchList.BeginUpdate;
// add high amount of items here
AdvSearchList.EndUpdate;
```

Categories

This is a collection of categories the list can hold. Categories can be filtered on. Each category in the categories list has an ID and this ID can be assigned to an item's CategoryID property.

The TCategoryItem properties are:

Caption: string
Text description of the category, used optionally to display in the category popup menu for the TAdvSearchEdit control or along with the items in the list.

Filter: Boolean
When true, all items with the category ID set to ID will be filtered, i.e. not displayed in the list.

ID: Integer
Sets the unique category ID.

Performing filtering

The TAdvSearchList.FilterCondition property controls the actual filtering of data in the list. It determines in what columns what data must match to retain or filter an item. The FilterCondition class has following properties:

AllColumns: Boolean
When true, a search for text match is done in all columns of the item

AutoSelect: Boolean
When true, the first matching item during a search will be automatically set in selected state in the list.

CaseSensitive: Boolean
Determines whether a search for a text match in the items is done with or without case-sensitivity

Categories: Boolean

When true, only items from the Categories collection where Filter = false are retained for display in the list.

CategoryItems: Boolean
When true, category items are also involved in the filtering process. When true, this means that category header items will also be automatically removed from list display when the category item does not match the search criteria. When false, the category items will always be retained and thus displayed.

Column: integer
Sets the column for which to check for a match. When AllColumns = false, a text search is only performed in the column with the index determined by this propery.

FilterData: TFilterData
Defines in what part of the column's item data to search for a match. This can be: the item's column caption, its description or both.

FilterType: TFilterType
This defines how the text match should be applied. The following options exist:
mText: item is retained when the text is found in any part of the text
mStartWord: item is retained when the text is found at the start of the item's column text
mEndWord: item is retained when the text is found at the end of the item's column text
mEntireWord: item is retained when there is a match on an entire word only with the item's column text

MaxCategoryItems: Integer
When different from -1, this sets an upper limit to the nr. of items to be retained for display from the

matching items. When the setting is -1, all matching items are displayed. When the setting is 10 for example, even when there are 100 matching items, only the first 10 will be displayed.

Text: string
Sets the text to search for.

When the filter conditions have been set or updated, simply call:

AdvSearchList.IUpdateFilter and a new filtering based on the new settings will be performed.

To remove a filter, call:

AdvSearchList.ClearFilter.

Appearance

Various settings are available to control the appearance of the items in the search list or the dropdown from the TAdvSearchEdit.

First there is the height of items. As there are two types of items: regular items and category items, there is a property to control the height for both types of items separately:

ItemHeight: sets the height of regular items

CategoryItemHeight: sets the height of category items

The Appearance class property has following properties to further control the appearance of the items:

BandColorEven: TColor
Sets the color for even items in the list when banding is enabled

BandColorOdd: TColor
Sets the color for odd items in the list when banding is enabled

Banding: Boolean
When true, BandColorEven / BandColorOdd colors are used as background color for even/odd items

CategoryColor: TColor
Sets the background color of category items

CategoryControlFont: Boolean
When true, category items use the control's font, when false, it uses the font set by CategoryFont

CategoryFont: TFont
Sets the font to be used by category items when CategoryControlFont = false

DescriptionControlFont: Boolean
When true, description text of items use the control's font, when false, it uses the font set by DescriptionFont

DescriptionFont: TFont
Sets the font to be used by the description in items when DescriptionControlFont = false

FilterCount: TFilterCount
When set to fcShow, the number of matching items for a category is displayed in the category item. The number is shown right aligned in the category item and formatted with format FilterCountFormat

FilterCountFont: TFont
Sets the font to use to show the filter count number right-aligned in the category items

FilterCountFormat: string
Holds the formatting specifier for the filter count number in the category item. %d designates the filter count number and the default value is set to '(%d)'.

HighlightColor: TColor
Sets the background color to use to highlight partial matching text in items. When the value is clNone, no background color is used.

HighlightFontStyle: TFontStyles
Sets the font style to use to highlight partial matching text in items.

HighlightTextColor: TColor
Sets the font color to use to highlight partial matching text in items. When the value is clNone, no font color is used.

ItemCategoryFont: TFont
Sets the font that is used to display the category name an item belongs to. The category name is derived from the Catogories collection's catergy Caption property based on the item's CategoryID property.

ItemCategoryFormat: string
Formatting specifier to display the category text with items when ShowItemCategory = true. %s designates the category name. The default ItemCategoryFormat is 'in %s'

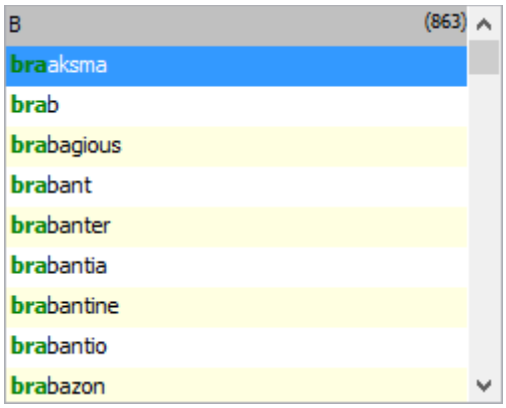
SelectionColor: TColor
Sets the background color of the selected item in the list

SelectionTextColor: TColor
Sets the font color of the selected item in the list

Example 1:

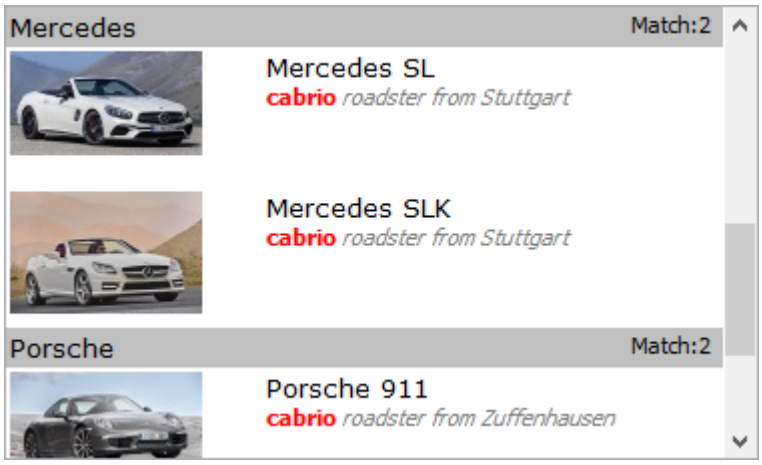
This is a search list with banding enabled. There is only one column. The number of items remaining

after filtering, the filtercount is displayed in the first category item. The filter type is set to mStartWord and HighLightColor is set to clGreen:



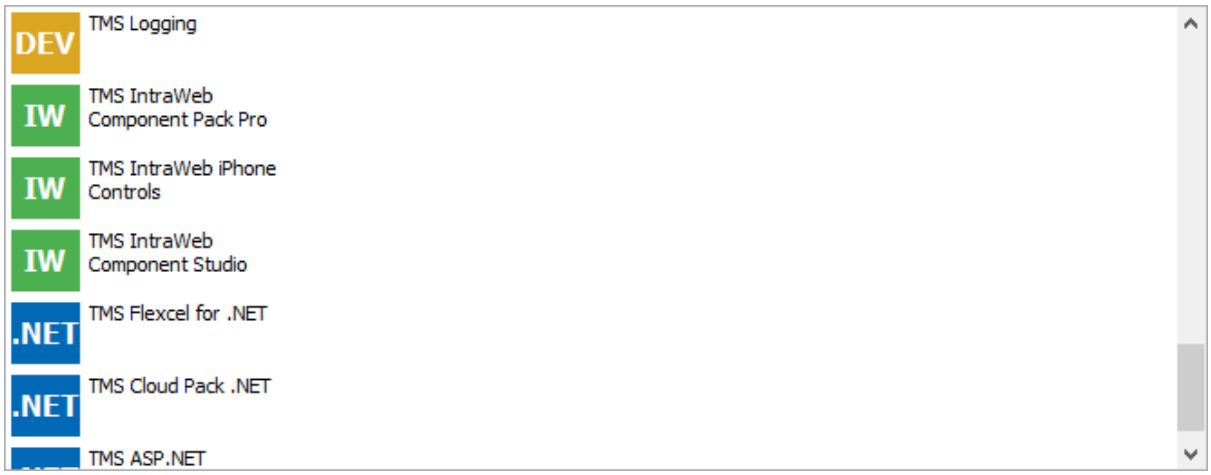
Example 2

This is a dual column search list. Filterdata is set to fdAll to search also in the description. FilterCountFormat is set to 'Match:%d' to indicate the number of found items in the category header.



Example 3

In this example, the items's first column shape is set to sRect and the text to 'DEV','IW','.NET',…



Events

Following events are specified to TAdvSearchList, TAdvSearchEdit:

OnDrawSearchItemColumn(Sender: TObject; AIndex, AColumn: Integer; ACanvas: TCanvas; ARect: TRect; ItemState: TItemState; var DefaultDraw: Boolean);

Event triggered to perform custom drawing of item's columns. It is triggered for each column within an item and the item is indicated by AIndex, the column by AColumn. Drawing can be done on the canvas ACanvas within rectangle ARect. The item's state, i.e. selected or not is returned and when DefaultDraw is set to false, no more default item drawing is done.

OnFilterItem(Sender: TObject; AItem: TSearchListItem; var Retain: Boolean);

Event triggered to perform custom filtering of items. When a new filter is started, this event is triggered for each item and to remove an item from the display, set the var parameter Retain = false.

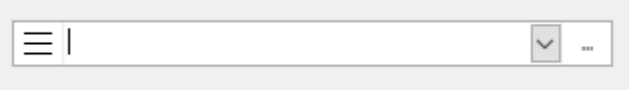
OnGetSearchItemColumn(Sender: TObject; AIndex, AColumn: Integer; var ACaption, ADescription: string; var AImageIndex: Integer);

Event triggered to set the item's column text, description and image dynamically so a virtual search list is possible. This event is triggered for each column for each displayed item and the text, description can be returned via the ACaption / ADescription parameter respectively. The item's column image index can be returned via AImageIndex.

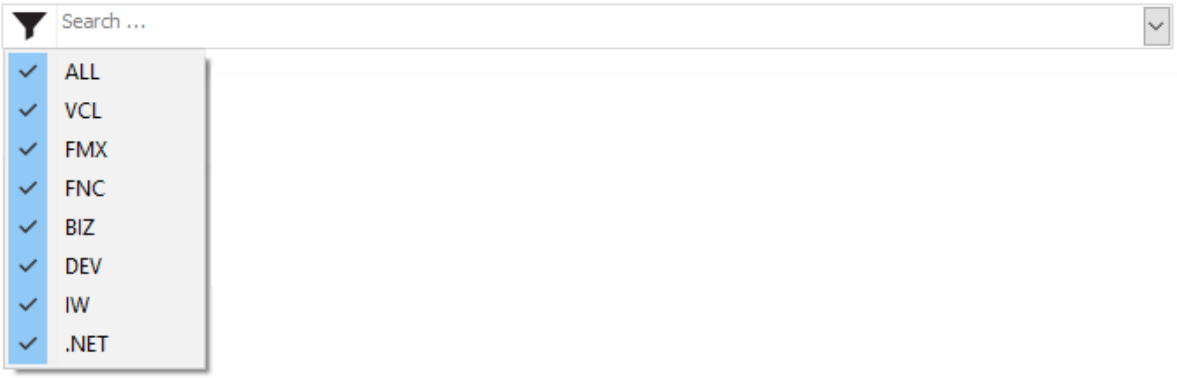
TAdvSearchEdit

TAdvSearchEdit is basically an edit control with two optional buttons that has a dropdown search list. While typing in the edit control, a search/filter is performed in the list and when pressing enter, the matching item is selected and its text is set in the edit control. It is critical here to set the FilterCondition.Column to the desired value as this determines from what column in the item the caption text will be used to update the edit control text when an item is selected.

The TAdvSearchEdit has two optional buttons, one left and one right from the edit control.



The left button is an optional button to select a category from for filtering. This button's properties can be set via TAdvSearchEdit.CategoryButton.



The CategoryButton can have a caption or a picture or both. It's width, appearance and visibility is set via TAdvSearchEdit.CategoryButton properties. Important to note is the property AdvSearchEdit.CategoryButton.PopupType. This defines the type of popup menu that is automatically displayed when the category button is clicked. By default, a popup menu is automatically created on the fly that contains a menu items all categories found under AdvSearchEdit.Categories. When the PopupType is set to pmCheck, a popup menu with check buttons is shown to allow to filter for multiple categories. When PopupType is set to pmRadio, a single category can be filtered on by selecting with a radio button.

The right button is an optional button from where a search can be started. This button's properties can be set via TAdvSearchEdit.SearchButton. It triggers the event OnSearchButtonClick when it is clicked.