TMS Component Pack
What's new guide

Jan, 2017

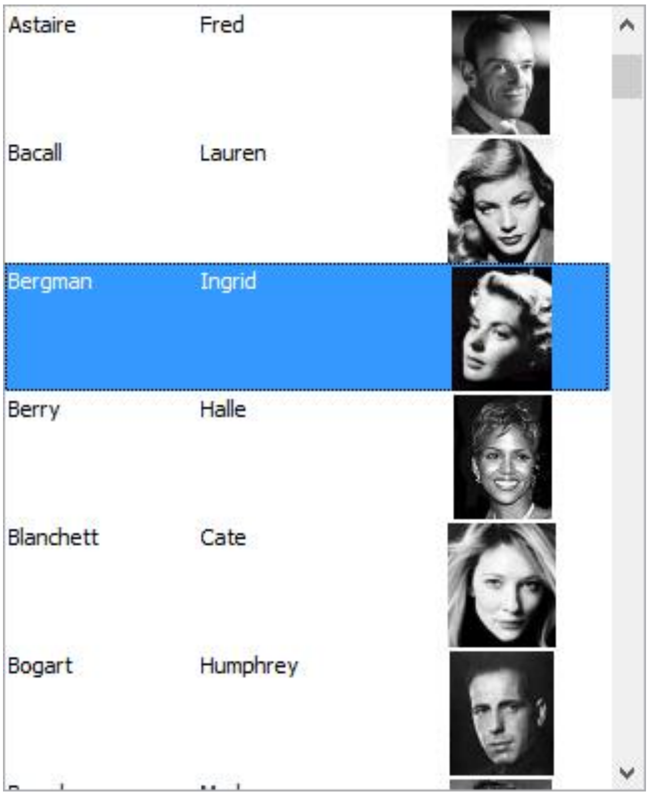
Web: http://www.tmssoftware.com
Email : info@tmssoftware.com

TMS Component Pack v8.5.0.0 - Jan 12, 2017

New: TDBAdvSearchList

DB-aware version of the TAdvSearchList component

The TDBAdvSearchList is a DB-aware of TAdvSearchList. This means that the data in the list in the different columns can be automatically filled from a connected dataset via a datasource. Set TDBAdvSearchList.DataSource and DBAdvSearchList.Columns[x].DataField to specify binding of values in the columns of the list.
When the DB field bound to a column is of the type TBlobFiled, the list control will try load the blob data as a picture.

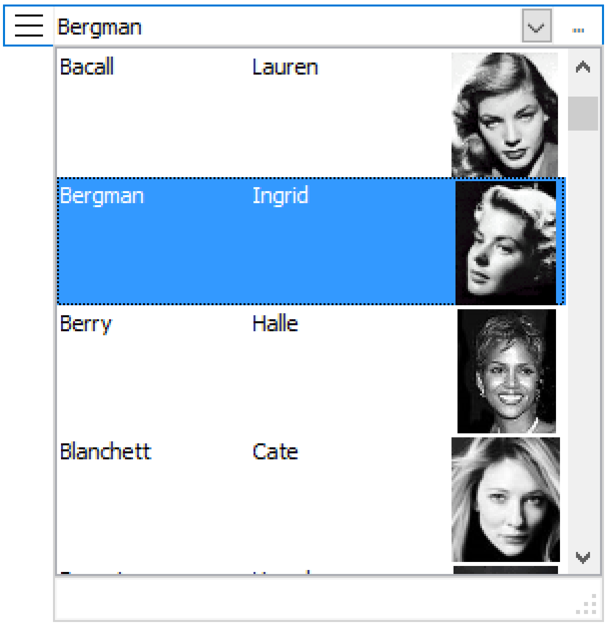


Astaire	Fred
Bacall	Lauren
Bergman	Ingrid
Berry	Halle
Blanchett	Cate
Bogart	Humphrey

New: TDBAdvSearchEdit

DB-aware version of the TAdvSearchEdit component

The TDBAdvSearchEdit is DB-aware version of the TAdvSearchEdit with respect to the value as well as with respect to the list of values. Set TDBAdvSearchEdit.DataSource and TDBAdvSearchEdit.DataField to define the DB-awareness of the value and use TDBAdvSearchEdit.ListSource in combination with TDBAdvSearchEdit.Columns[x].DataField to specify binding of values in the columns of the list.



Bergman

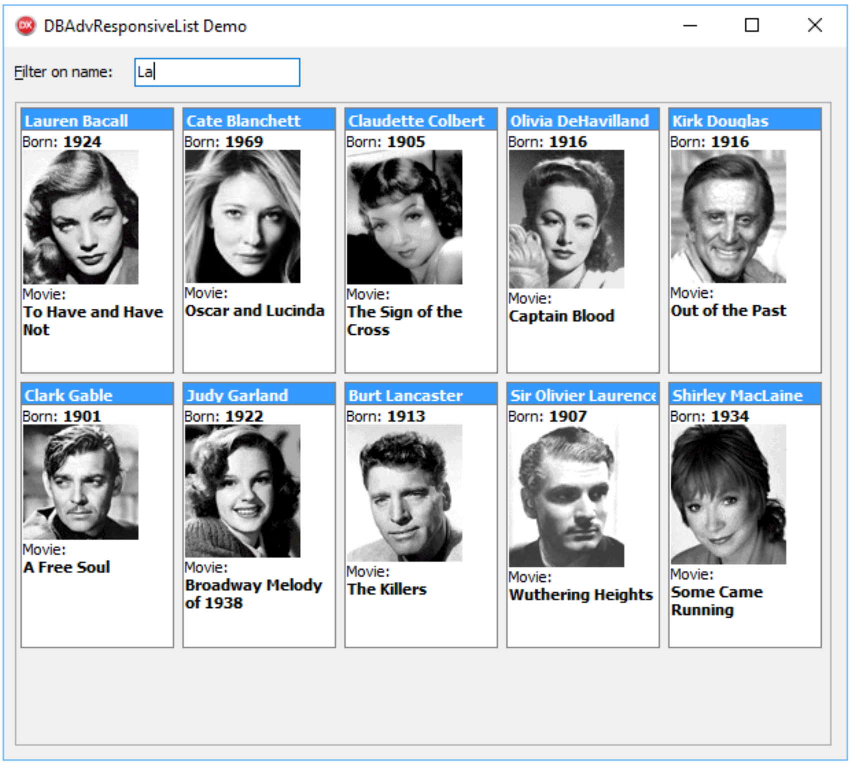
Bacall	Lauren	
Bergman	Ingrid	
Berry	Halle	
Blanchett	Cate	

New: TDBAdvResponsiveList

DB-aware version of the TAdvResponsiveList component

The TDBAdvResponsiveList is a DB-aware version of the TAdvResponsiveList. Header, content and footer of responsive list items can be loaded automatically from a dataset connected via a datasource. Set the datasource via TDBAdvResponsiveList.DataSource. The binding of values to items is done via item templates. The template placeholders are defined as (#FIELDNAME) and are dynamically rendered when the dataset is activated. Memo fields can be used as well as picture blob fields. For picture blob fields, specify the template value to render as picture as:

<IMG src=”(#BLOBFIELDNAME)”>



DBAdvResponsiveList Demo

Filter on name: La

Lauren Bacall
Born: 1924
Movie:
To Have and Have Not

Cate Blanchett
Born: 1969
Movie:
Oscar and Lucinda

Claudette Colbert
Born: 1905
Movie:
The Sign of the Cross

Olivia DeHavilland
Born: 1916
Movie:
Captain Blood

Kirk Douglas
Born: 1916
Movie:
Out of the Past

Clark Gable
Born: 1901
Movie:
A Free Soul

Judy Garland
Born: 1922
Movie:
Broadway Melody of 1938

Burt Lancaster
Born: 1913
Movie:
The Killers

Sir Olivier Laurence
Born: 1907
Movie:
Wuthering Heights

Shirley MacLaine
Born: 1934
Movie:
Some Came Running

New: TDBAdvOfficeComboBox

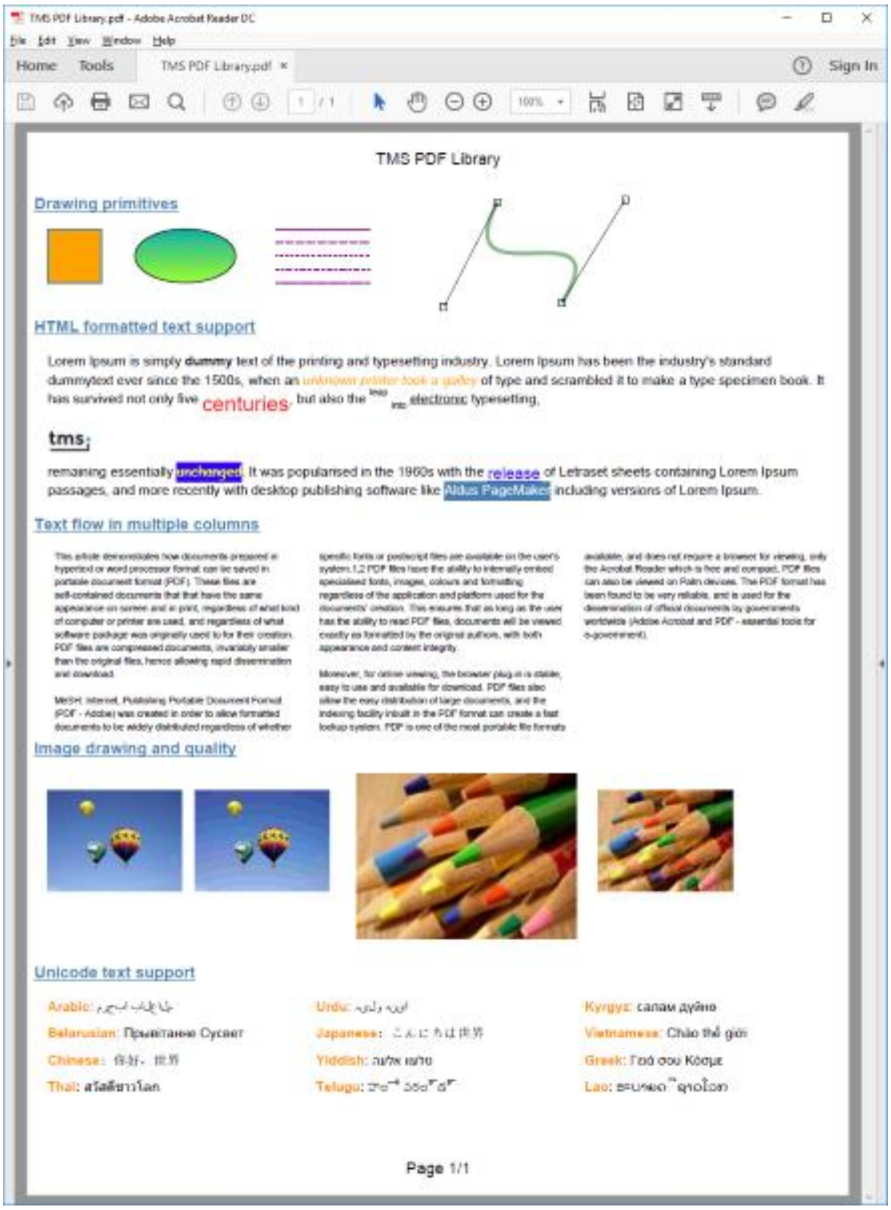
DB-aware version of the TAdvOfficeComboBox component

The combobox can be data-bound with respect to value as well as with respect to list values. Use TDBAdvOfficeComboBox.DataSource/ TDBAdvOfficeComboBox.DataField to define the DB binding for the combobox value and use TDBAdvOfficeComboBox.ListSource / TDBAdvOfficeComboBox.ListField to define the DB binding for the list values.

New: TAdvPDFLib

Standalone PDF library to generate PDF files easily on-the-fly.

Library that features PDF file generation with text, formatted text, rectangles, lines, polygons, images, … header, footer access and custom drawing.


TMS PDF Library.pdf - Adobe Acrobat Reader DC
File Edit View Window Help
Home Tools TMS PDF Library.pdf ×
Sign In
1 / 1
100%

TMS PDF Library

Drawing primitives

HTML formatted text support

Lorem Ipsum is simply **dummy** text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummytext ever since the 1500s, when an *unknown printer took a galley* of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting,

tms

remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Text flow in multiple columns

This article demonstrates how documents prepared in hypertext or word processor format can be saved in portable document format (PDF). These files are self-contained documents that that have the same appearance on screen and in print, regardless of what kind of computer or printer are used, and regardless of what software package was originally used to for their creation. PDF files are compressed documents, invariably smaller than the original files, hence allowing rapid dissemination and download.

MeSH: Internet, Publishing Portable Document Format (PDF - Adobe) was created in order to allow formatted documents to be widely distributed regardless of whether specific fonts or postscript files are available on the user's system.1,2 PDF files have the ability to internally embed specialised fonts, images, colours and formatting regardless of the application and platform used for the documents' creation. This ensures that as long as the user has the ability to read PDF files, documents will be viewed exactly as formatted by the original authors, with both appearance and content integrity.

Moreover, for online viewing, the browser plug-in is stable, easy to use and available for download. PDF files also allow the easy distribution of large documents, and the indexing facility inbuilt in the PDF format can create a fast lookup system. PDF is one of the most portable file formats available, and does not require a browser for viewing, only the Acrobat Reader which is free and compact. PDF files can also be viewed on Palm devices. The PDF format has been found to be very reliable, and is used for the dissemination of official documents by governments worldwide (Adobe Acrobat and PDF - essential tools for e-government).

Image drawing and quality

Unicode text support

Arabic: مرحبا بالعالم	Urdu: ہیلو ورلڈ	Kyrgyz: салам дүйнө
Belarusian: Прывітанне Сусвет	Japanese: こんにちは世界	Vietnamese: Chào thế giới
Chinese: 你好，世界	Yiddish: העלא וועלט	Greek: Γεά σου Κόσμε
Thai: สวัสดีชาวโลก	Telugu: హలో వరల్డ్	Lao: ສະບາຍດີຊາວໂລກ

Page 1/1

There is a new developers guide specifically covering the new PDF generation library.

New: TAdvPDFImageBook

Class that can on-the-fly generate PDF files from a list of images

TAdvResponsiveList

- New: HeaderTemplate capability added

Now also for the item header, a template can be specified. The template placeholders are just like for the content specified as (#VALUENAME). The template is rendered by using the TResponsiveListItem.Values['VALUENAME'] value and have specifiers like (#VALUENAME) replaced by these values.

- New: FooterTemplate capability added

Same as the header template but for the optional footer of the item

- New: Appearance.ItemContentMargin added

New property to specify the margin from left/right and top/bottom of the content with respect to the outline of the item.

- New: Filter capability

Items in the TAdvResponsiveList can now be easily filtered. The filter condition is set via TAdvResponsiveList.FilterCondition. Following properties are available:

FilterCondition.CaseSensitive: Boolean : when true, a case-sensitive match is performed

FilterCondition.FilterData: TResponsiveListItemFilterDataSet : defines on what data to filter : the header, the content or the footer.

```
TResponsiveListItemFilterData = (fdContent, fdHeader, fdFooter);
TResponsiveListItemFilterDataSet = set of TResponsiveListItemFilterData;
```

FilterCondition.FilterType: TResponsiveListItemFilterType : defines what match method to use:

```
TResponsiveListItemFilterType = (mText, mEntireWord, mStartWord, mEndWord);
```

mText: match of any part of the text
mEntireWord: match of text respecting word boundaries
mStartWord: match must be at start of a word
mEndWord: match must be at end of a word

FilterCondition.Text : string : holds the text to perform a match against

The filter operation is started by setting the filter condition and then call:

```
TAdvSearchList.UpdateFilter;
```

The filter operation is cancelled by calling:

```
TAdvSearchList.ClearFilter;
```

In addition to the filter condition, an event OnItemFilter() is also triggered for each item's filter match check and as such, via this event also custom filtering can be implemented. When this event is triggered, set the var parameter Retain to TRUE if the item needs to remain in the list after filtering.

TAdvStringGrid

- New: XYRTFOffset public property added to control offset of only RTF text

The property grid.XYOffset controls the offset from top/left coordinate of the cell to the text and the property grid.XYRTFOffset controls this same offset but for RTF rendered cell text

- New: Event OnBeforeFilter added

This event is triggered just before filtering from the grid in response to setting filter conditions from the filterbar will happen. At the moment OnBeforeFilter() is triggered, the filter conditions from the filterbar are all defined in the collection grid.Filter. This means it is possible to programmatically dynamically change/extend the filter conditions just before these are applied.

- New: Grid.Navigation.MoveRowOnScroll property added

When true, the focused row is moved and remains within the visible range of rows in the grid.

TAdvRichEditor

- New: OnDropFile event added

Event triggered when a file is dropped on the TAdvRichEditor. This event passes the name of the file that is about to be dropped and with the var parameter Allow, the file drop can be dynamically blocked.

- New: Public property SpaceAsNbSp: boolean added

When true, spaces in the text are exported as HTML special characters &nbsp; and otherwise as regular space characters

TAdvSearchList

- New: Method UnSelect added

Call AdvSearchList to programmatically unselect any previously selected items

- New: WordWrapping & Trimming properties added at column level

Now it is possible to specify for each column in the searchlist whether the text will be rendered with wordwrapping or not and what type of trimming needs to be applied.

TAdvSearchEdit

- New: Property ItemIndex added at TAdvSearchEdit level

Public property that allows to get & set the selected item within the search edit dropdown list

- New: Exposed Images property to set an imagelist

Allows to set the imagelist for images that will be used in the search edit dropdown list

TAdvEditBtn

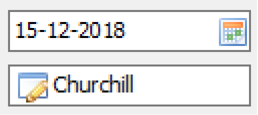
- New: PictureHot, PictureDown, PictureDisabled properties added

Allows to set (as PNG, BMP, GIF, JPG file) the pictures to be used for the embedded button for the normal, hot, disabled and down state

- New: ButtonAlign property added

Specifies whether the button is at the left or right side in the edit control

- New: ButtonTransparent property added
When true, the button in the edit control is transparent



TAdvShapeButton

- New: Property ShowDropPicture: boolean added

When false, the dropdown picture for a dropdown button type will not be displayed. The default is that the dropdown picture (arrow down by default) will be displayed.

TAdvDirectoryTreeView

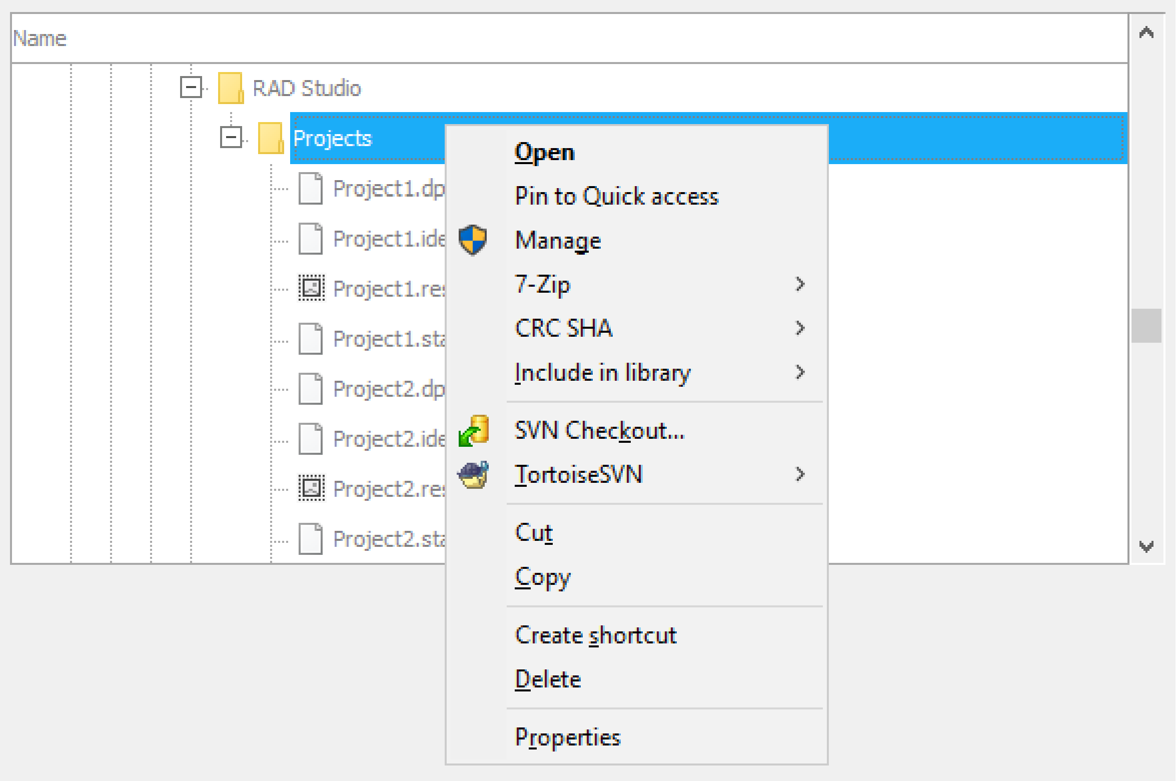
- New: SortOrder: TAdvDirectoryTreeViewSortOrder to set sorting order to files first, folders first

```
  TAdvDirectoryTreeViewSortOrder = (tvsoNone, tvsoFilesFirst,
tvsoDirectoriesFirst);
```

Defines whether to display first files in the treeview, first directories or use the default order of returned files and directories by the shell.

- New: SystemContextMenu: Boolean;

When this property is true, the property shell context menu will be shown on right-click of a file/folder in the treeview.



TAdvGauge

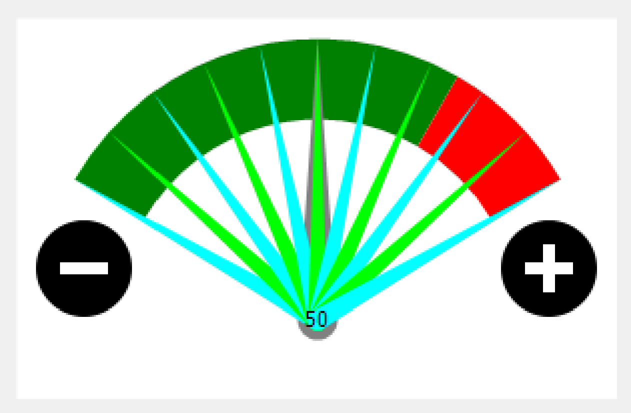
- New: ExtraNeedles: TNeedleItems

This is an extra collection to add as many needles as desired to the gauge. The collection consists of TNeedleItem objects via which the needle color, position and width can be set.

Example:

```
var
  it: TNeedleItem;
  I: Integer;
begin
  for I := 0 to 10 do
  begin
    it := AdvGauge1.ExtraNeedles.Add;
    it.Position := i * 10;
    if odd(i) then
      it.Color := clLime
    else
      it.Color := clAqua;
  end;
end;
```

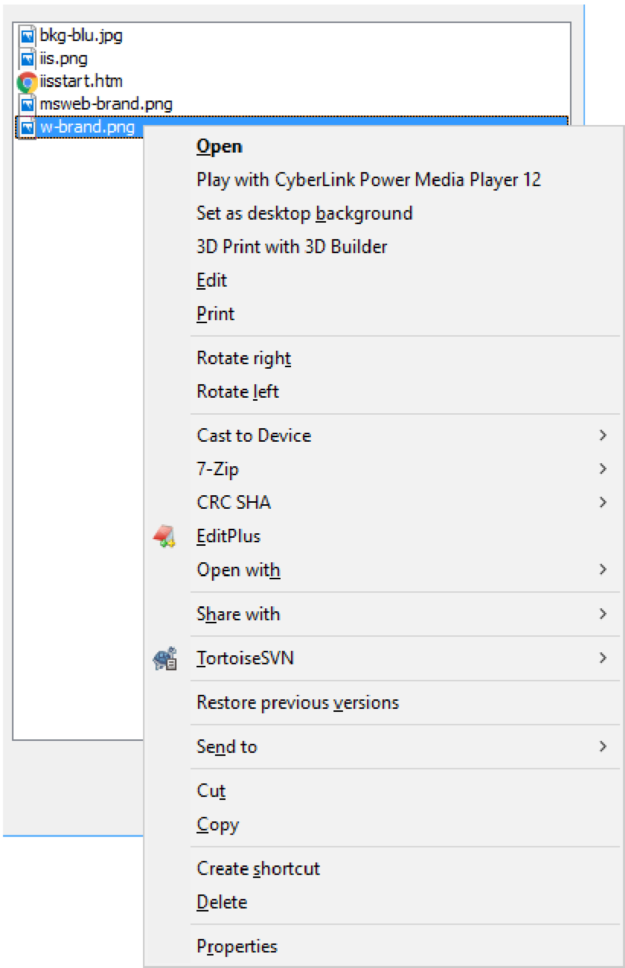
This code adds 11 needles and this results in:



TFileListBoxEx, TCheckFileListBoxEx

- New: property SystemContextMenu: Boolean

When this property is true, the property shell context menu will be shown on right-click of a file in the TFileListBoxEx or TCheckFileListBoxEx



TMS Component Pack v8.4.2.0 - Nov 29, 2016

TAdvStringGrid

- New: Event OnFilterEditDone added, triggered after new filtering condition was set

Event triggered when the filtering operation was performed after the filter conditions were updated from the automatic filter edit bar

- New: Public property DoAutoEditFilter added to control to apply filtering automatically after edit

When true, filtering will be automatically and directly updated when a value in the filter bar editors changes

TAdvSearchList

- New: Property WheelIncrement: integer;

Sets the number of rows a mouse wheel up or mouse wheel down event the active row will move up or move down

TMS Component Pack v8.4.1.0 - Nov 15, 2016

TAdvStringGrid

- New: Sort type ssVarDate added for more flexible date recognition

With the ssVarDate sort type, date formats supported by VarToDateTime are also supported

TAdvRichEditor

- New: TAdvRichEditorHTMLIO.SpaceAsNbSp: boolean property added

When true, spaces are exported to HTML as special characters &nbsp; otherwise as regular spaces

TAdvSearchEdit

- New: Exposed TabOrder property

- New: TAdvSearchEdit.SearchList exposed

Provides access to the searchlist control in the dropdown

- New: FocusColor, FocusFontColor, FocusBorder, FocusBorderColor properties added

- New: DisabledColor property added

TAdvTaskDialog

- New: Public properties DefaultButtonHeight, DefaultButtonWidth

Allows to defines the default sizes for the buttons in the task dialog

THTMListBox

- New: Exposed OnMouseLeave, OnMouseEnter events

TAdvLabelEdit, TAdvSpinEdit, TAdvDBComboBox

- New: Exposed StyleElements property

TAdvTreeView

- New: FindNodeByRow and GetTotalNodeCount functions

New functions to find a node by row index and to get the total number of nodes

TAdvSmoothImageListBox

- New: ZoomOutOnDblClick property added

When true, a zoomed in image in the list will be automatically return to default state when double-clicked.

TMS Component Pack v8.4.0.0 - Oct 25, 2016

TAdvStringGrid

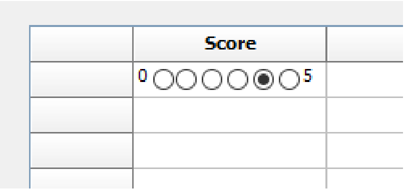
- New: property grid.ControlLook.RadioRange: Boolean

Settings this property to true allows a shortened display of a radiogroup where the first radiobutton text value is shown left from the radiobutton and the last one right from the button. This allows to have only the minimum text left and right of the radiobutton group.

Code snippet:

```
var
  sl: TStringList;
begin
  AdvStringGrid1.Cells[1,0] :='Score';
  AdvStringGrid1.Alignments[1,0] := taCenter;
  sl := TStringList.Create;
  sl.Add('0');
  sl.Add('');
  sl.Add('');
  sl.Add('');
  sl.Add('');
  sl.Add('5');
  AdvStringGrid1.ControlLook.RadioRange := true;
  AdvStringGrid1.ColWidths[1] := 120;
  AdvStringGrid1.AddRadio(1,1,1,0,sl);
end;
```

This results in:



- New: event OnOleDropFiles added to handle OLE drag & drop of multiple files

This event is triggered in case multiple files are dropped on the grid via OLE drag & drop. The event signature is:

```
OnOleDropFiles(Sender: TObject; ARow, ACol: Integer; Files: TStrings);
```

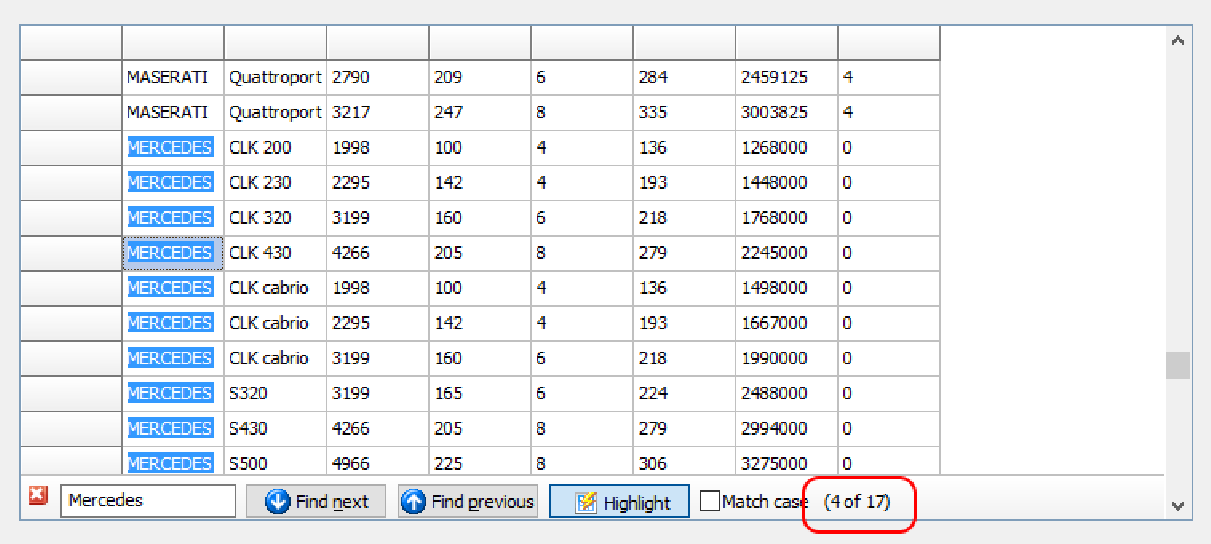
It indicates the cell where the drop happens and returns the list of filenames

- New: Search option fnIgnoreAccents added

With this new option, when it is added to the search parameters, the search does not take accented characters in account. I.e. if a cell contains the value “élève”, it will be a match when a search is done for “eleve”.

- New: grid.SearchFooter.ShowResults: Boolean property / grid.SearchFooter.ResultFormat: string property added

When true, the number of matches for a search from the search footer is displayed in the search footer. The SearchFooter.ResultFormat can specify the formatting string for this number.



	MASERATI	Quattroport	2790	209	6	284	2459125	4
	MASERATI	Quattroport	3217	247	8	335	3003825	4
	MERCEDES	CLK 200	1998	100	4	136	1268000	0
	MERCEDES	CLK 230	2295	142	4	193	1448000	0
	MERCEDES	CLK 320	3199	160	6	218	1768000	0
	MERCEDES	CLK 430	4266	205	8	279	2245000	0
	MERCEDES	CLK cabrio	1998	100	4	136	1498000	0
	MERCEDES	CLK cabrio	2295	142	4	193	1667000	0
	MERCEDES	CLK cabrio	3199	160	6	218	1990000	0
	MERCEDES	S320	3199	165	6	224	2488000	0
	MERCEDES	S430	4266	205	8	279	2994000	0
	MERCEDES	S500	4966	225	8	306	3275000	0

Mercedes

- New: grid.SearchFooter.IgnoreAccents: Boolean property added

When true, the search from the search footer is done without taking accented characters in account.

- New : Public property grid.PrintSettings.ScaleFooter, grid.PrintSettings.ScaleHeader added

By default, this setting is true. This means that when a fit to page is happening and the font needs to be scaled down, the header and footer text will also scale down. When grid.PrintSettings.ScaleHeader = false, the header size will always be the same, regardless of scaling on the grid itself.

- New: Event OnFilterCheckMouseDown event added

This event is triggered when a mouse down event happens on the filter checklist when grid.FilterDropDownCheck is set.

- New: grid.HyperLink[col,row]: string added

This returns the hyperlink URL itself in case a grid cell has a hyperlink. In case the cell contains HTML formatted text with a hyperlink, it returns the first hyperlink URL found.

Example:

```
grid.Cells[1,1] := 'Click <a href=http://www.tmssoftware.com>here</a> to
visit website';

ShowMessage(grid.HyperLink[1,1]);
```

This will return: 'http://www.tmssoftware.com'

- New: function grid.HasHyperLink(col,row): Boolean added

This function returns true when a grid cell has a hyperlink.

- New: TAdvPreviewDialogDialogResizable added

Default this is true and the print preview dialog can be resized. Set this to false, to make the preview dialog stick to the fixed preset size.

- New: Optional parameter IgnoreHTML added in grid.FindMulti() method

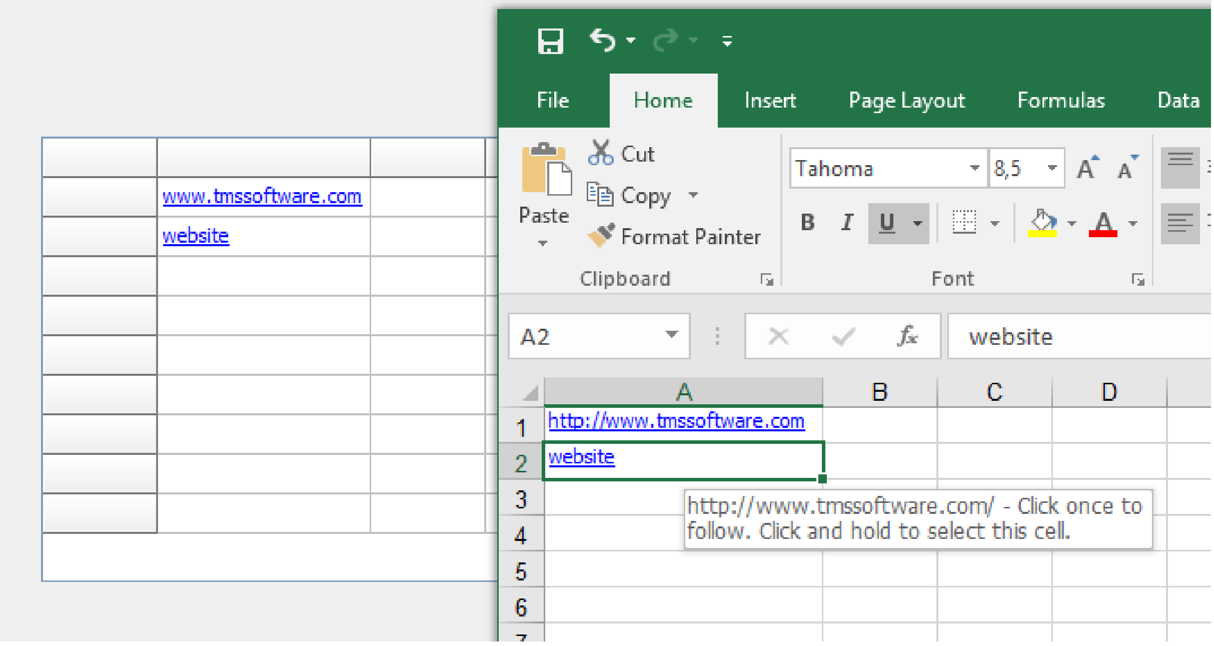
A new parameter was added to the grid.FindMulti() method. When the last parameter IgnoreHTML is set to true, the possible HTML tags in HTML formatted cells are ignored.

- New: Hyperlink export in TAdvGridExcelIO

A new option is added in TAdvGridExcelIO that lets it export hyperlinks in grid cells as real hyperlinks in the Excel file generated. When a cell text contains a hyperlink value, this value is directly exported as hyperlink to the Excel cell. When the grid cell text contains a hyperlink via HTML formatted text, the hyperlink text is exported itself with the URL retrieved from the href attribute. When a cell text would contain multiple hyperlinks via HTML formatted text, only the first hyperlink will be exported as an Excel cell can only contain one hyperlink.

This is illustrated with following sample:

```
begin
  AdvStringGrid1.Cells[1,1] := 'http://www.tmssoftware.com';
  AdvStringGrid1.Cells[1,2] := '<a
href="http://www.tmssoftware.com">website</a>';
  AdvStringGrid1.URLShow := true;
  AdvStringGrid1.AutoSizeCol(1);
  AdvGridExcelio1.XLSExport('.\gridlink.xls');
end;
```



www.tmssoftware.com

website

	A	B	C	D
1	http://www.tmssoftware.com			
2	website			
3				
4				
5				
6				

http://www.tmssoftware.com/ - Click once to follow. Click and hold to select this cell.

TAdvGraphicCheckLabel

- New: property ShowFocus: Boolean added

When true, a dotted focus border is shown around the control when it has focus.

- New: property ReturnIsTab: Boolean added

When true, pressing the return key while the control has focus moves the focus to the next or previous control (depending on the shift key)

- New: property Transparent: Boolean added

Sets the control transparent or not.

TAdvOfficeTabSet

- New: public property ScrollMin: integer, ScrollMax: integer added

These properties return the minimum and maximum scroll position that can be programmatically set respectively.

TAdvDBComboBox

- New: property ReturnIsTab: Boolean

When true, pressing the return key while the control has focus moves the focus to the next or previous control (depending on the shift key)

TAdvMultiInputQueryDialog

- New: Clear button capability added, OnClearClick event added

Set ShowClearButton = true to have a new button that will clear the entered values

- New: Default button capability added, OnDefaultClick event added

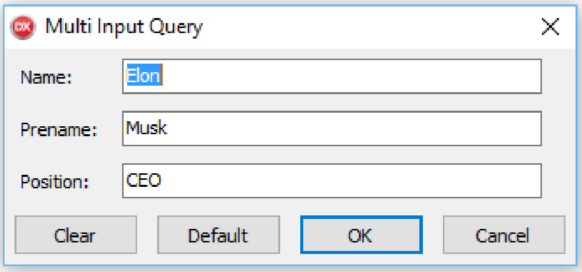
Set ShowDefaultButton = true that will reset input values to the preset default values when clicked

- New: ButtonsAlignment property added

Buttons can be aligned left or right from the dialog

- New: CaptionOK, CaptionCancel, CaptionClear, CaptionDefault properties added

Properties to customize the captions of the buttons


Multi Input Query

Name: Elon

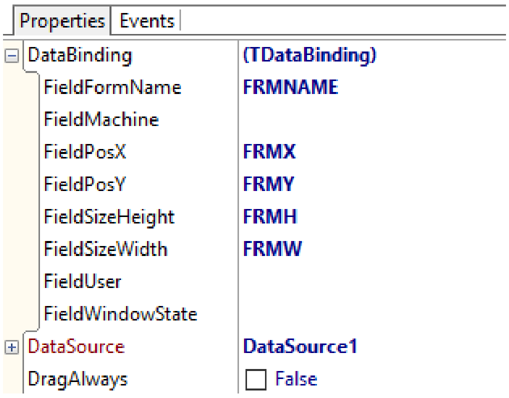
Prename: Musk

Position: CEO

TFormSize

- New: Capability to persist settings in a dataset added

You can connect a dataset via a datasource to the TFormSize component to allow to persist the form position and size settings to a database. The FormSize.DataBinding property allows to define via which dataset field the data is persisted. Not all fields need to be defined but the minimum needed will be FieldFormName, FieldPosX, FieldPosY, FieldSizeWidth, FieldSizeHeight



Properties	Events
⊟ DataBinding	(TDataBinding)
FieldFormName	FRMNAME
FieldMachine	
FieldPosX	FRMX
FieldPosY	FRMY
FieldSizeHeight	FRMH
FieldSizeWidth	FRMW
FieldUser	
FieldWindowState	
⊞ DataSource	DataSource1
DragAlways	□ False

TCheckListEdit

- New: Capability to have an attached label added

Just like with the other edit controls, TCheckListEdit can now also have an attached label. Set the caption with CheckListEdit.LabelCaption, the font with CheckListEdit.LabelFont.

TAdvProgressBar

- New: methods StepUp / StepDown added

Two methods are added to increase or decrease the progressbar position

TAdvGlowButton

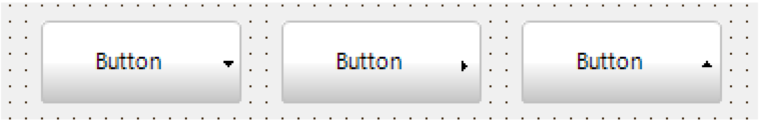
- New: DropDownDirection ddUp added

This permits to have an upwards arrow in the dropdown button area

- New: DownPicture property added

Via this property, a separate image can be defined for the button down state

The three types of dropdown buttons:



TAdvListView

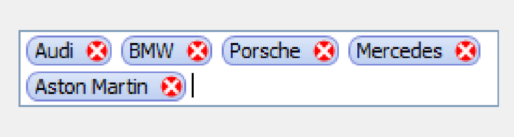
- New: event OnSaveValue added

Now it is possible to override the values that will be written to any of the file formats TAdvListView can export to by using the OnSaveValue event. OnSaveValue is triggered for each item and each subitem within an item. The ItemIndex parameter indicates the item, the SubItemIndex is -1 for the item caption value itself or is a value between 0 and the number of SubItems to refer to the SubItem value index.

TAdvListEditor

- New: property ShowDeleteButton added

When ShowDeleteButton is set true, the items in the list editor get an extra small delete button from where the items can be instantly deleted with a mouse click



TAdvDBLookupComboBox

- New: Label hint capability added

When the label is too small to show the full label text, the label text is shown with ellipsis and the full label text is shown as a hint when the mouse hovers the label.

TAdvMetroButton

- New: DropDown, DropDownMenu properties added

THTMLCheckBox

- New: property AllowGrayed added

When AllowGrayed is set true, the checkbox state toggles through cbChecked, cbGrayed, cbUnChecked. When not, only states cbChecked, cbUnChecked are possible.

TAdvRatingGrid

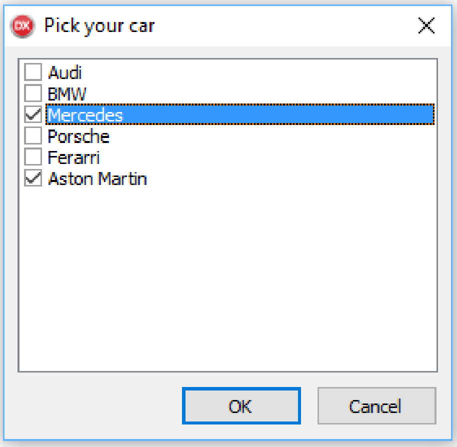
- New: RatingCategory.Visible property added

A category can be (temporarily) made invisible by setting Visible = false.

TPickDialog

- New: ListType property added to select between ltList, ltCheckList

Select with PickDialog.ListType if a checklistbox or listbox is used to show the list of possible values. When ListType is set to ltCheckList, the PickDialog will always behave as a multi select PickDialog.

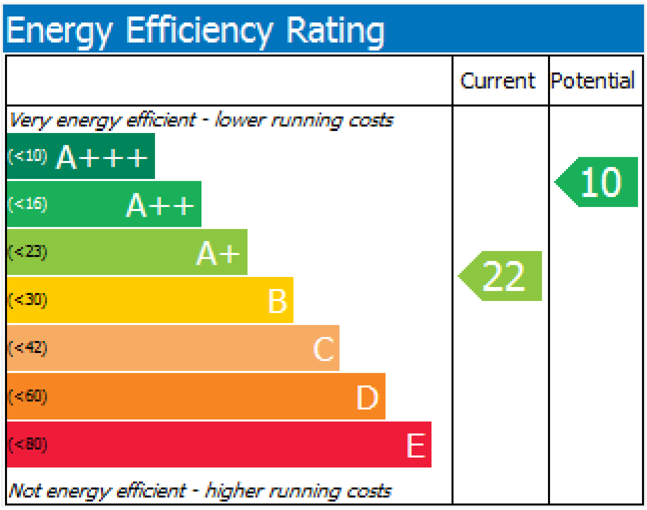


☐ Audi
☐ BMW
☑ Mercedes
☐ Porsche
☐ Ferarri
☑ Aston Martin

TSPBEPC

- New: ChartType ctEPC2010EEI added

When ctEPC2010EEI chart type is chosen, the new rating classes A+, A++,A+++ are shown.



TAdvInputTaskDialog

- New: AutoClose, AutoCloseTimeOut property added

When AutoClose is set to true, the dialog closes itself after the number of milliseconds set by the AutoCloseTimeOut value.

TEXEInfo

- New: property LinkerDateTime: TDateTime added

This returns the linke timestamp stored in the application EXE header

TAdvDateTimePicker

- New: event OnCheckClick added

The event is triggered when the checkbox is clicked that is displayed in the date part when the property ShowCheckBox is set to true.

TAdvToolBarPager

- New: drag & drop events exposed

The events OnDragDrop, OnDragOver, OnEndDrag, OnStartDrag were added

TAdvIPEdit

- New: properties IPv4Address: Cardinal and IPv6Address: TIPv6Address added

Now it is also possible to get and set the IP address value as a number instead of text. In IPv4 mode, the IP address is get or set via a 32bit number. When IPv6 is chosen, the IP address is set via a TIPv6Address record that is a record of 2 64bit values.

TAdvMaskEdit

- New: property DefaultHandling: Boolean added in TAdvMaskEdit

When DefaultHandling is true, the return key or ESC key will be handled by TAdvMaskEdit instead of the button that might be set as default button on the form.

TAdvOfficePager

- New: property Text: string added

This sets the text shown when there is no TAdvOfficePage in TAdvOfficePager

- New: property TAdvOfficePage.Text: string added

This sets a text that is displayed on the TAdvOfficePage in the TAdvOfficePager

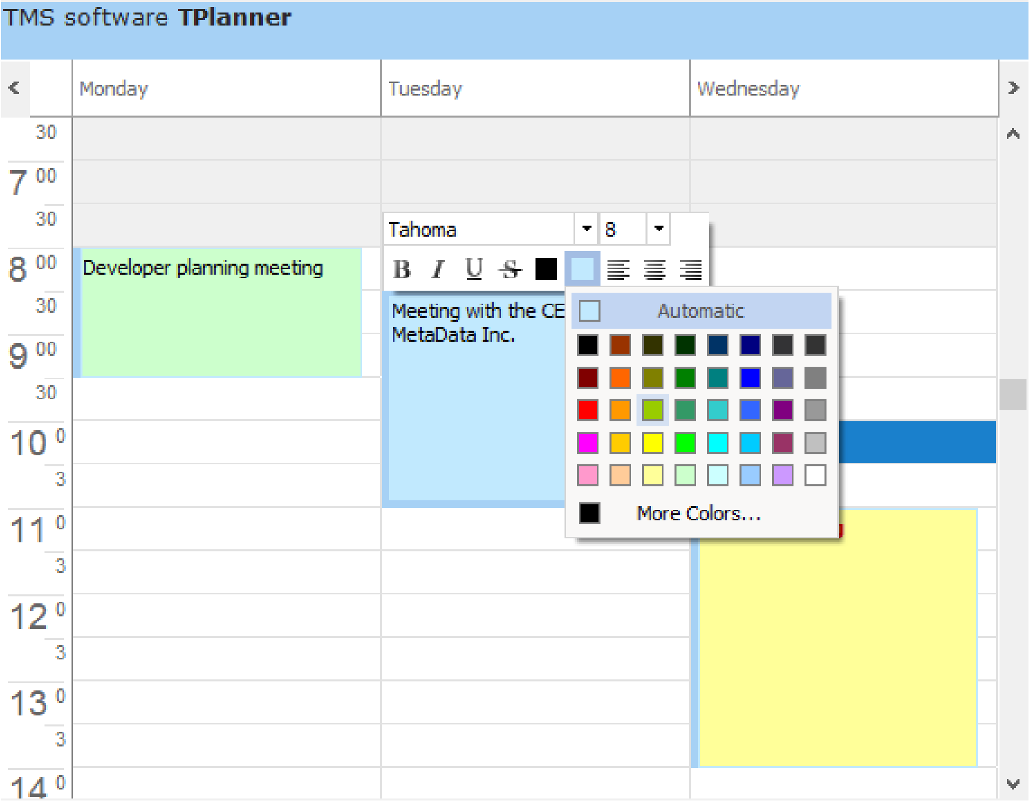
TPlanner

- New: property TPlannerItem.Ellipsis: Boolean added

When TPlannerItem.Ellipsis is set to true and TPlannerItem.WordWrap is set to false, the TPlannerItem text is shown with ellipsis when it does not fit in the TPlannerItem area, otherwise it is cut off at the border.

-New: PlannerPopupToolBar

The TPlanner has a new property PopupToolBar. A TPlannerPopupToolBar component can be assigned to this property. When it is assigned, a popup toolbar will appear next to a TPlannerItem when the mouse hovers over a selected TPlannerItem. From this popup toolbar, the font, font attributes and color of the TPlannerItem text can be directly set.



TAdvEditBtn

- New: property Picture: TGDIPicture added

With the new Picture property, JPG,GIF,PNG image formats can be used as the picture that is shown on the button in TAdvEditBtn.

TAdvSplitter

- New: Public property State: TSplitterState added

When the TAdvSplitter.DblClickAction is set to dbaOpenClose, a double click on the splitter toggles between an open & closed state. The public readonly property state returns whether the splitter is in open or closed state or in an unknown state (somewhere in between open or closed)

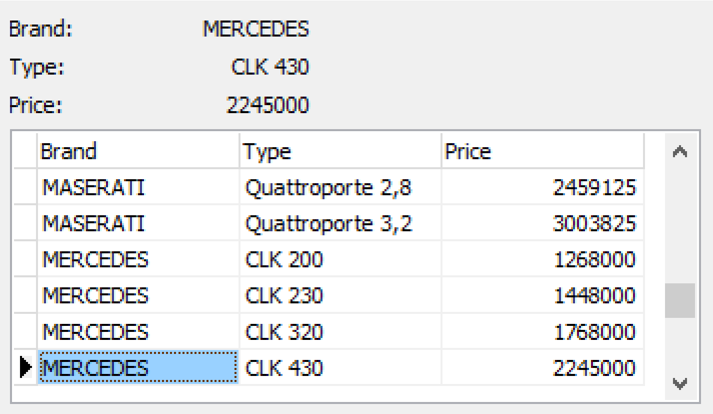
TSplitterState = (stUnknown, stOpen, stClose);

TAdvDBDataLabel

- New component: DB-aware version of TAdvDataLabel

TAdvDataLabel is a dual label component that can display different value types in specific defined formats. The data value is retrieved via TAdvDBDataLabel.DataSource / TAdvDBDataLabel.DataField.

Here, three TAdvDBDataLabel controls are used to show the DB values as label where the field name is automatically also displayed and the field value is shown with right alignment as well.



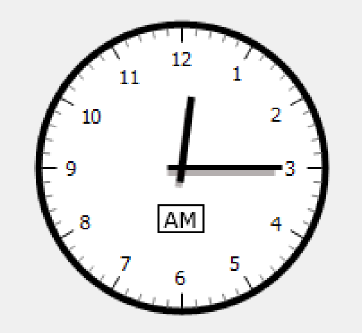
Brand: MERCEDES
Type: CLK 430
Price: 2245000

Brand	Type	Price
MASERATI	Quattroporte 2,8	2459125
MASERATI	Quattroporte 3,2	3003825
MERCEDES	CLK 200	1268000
MERCEDES	CLK 230	1448000
MERCEDES	CLK 320	1768000
MERCEDES	CLK 430	2245000

TAdvWatch

- New: property Resolution: integer (1..30) added

The new resolution determines what the resolution can be of the minutes for selection. When Resolution is set to 5, only minute values 0,5,10,15… can be selected by clicking the watch outer circle. Default Resolution value is 1 to allow to pick any minute value.



TAdvTimePickerDropDown

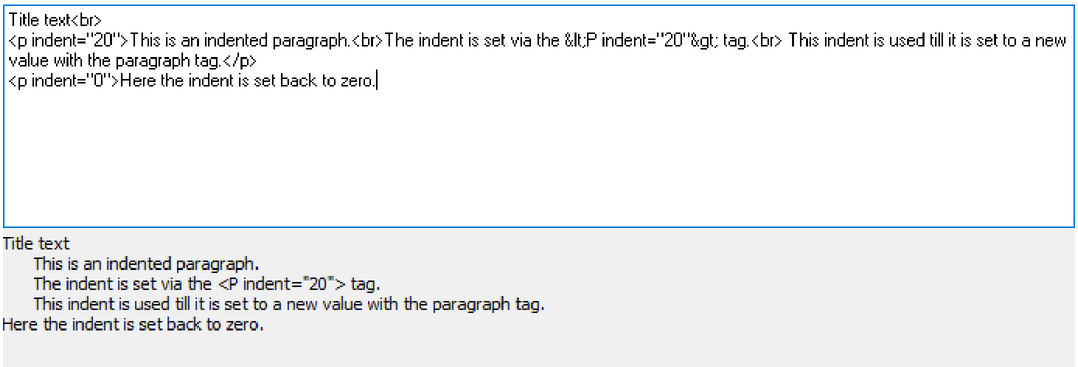
- New: property Resolution added

The new resolution determines what the resolution can be of the minutes for selection. When Resolution is set to 5, only minute values 0,5,10,15… can be selected by clicking the watch outer circle. Default Resolution value is 1 to allow to pick any minute value.

HTML Controls

- New: paragraph indent attribute added

The <P indent="value"> indent attribute allows to define a paragraph indent that is kept till it is reset.



```
Title text<br>
<p indent="20">This is an indented paragraph.<br>The indent is set via the &lt;P indent="20"&gt; tag.<br> This indent is used till it is set to a new
value with the paragraph tag.</p>
<p indent="0">Here the indent is set back to zero.
```

```
Title text
    This is an indented paragraph.
    The indent is set via the <P indent="20"> tag.
    This indent is used till it is set to a new value with the paragraph tag.
Here the indent is set back to zero.
```

TWebUpdate

- New: Capability added to show HTML formatted text in the what's new dialog

When the text for the what's new dialog in TWebUpdate contains HTML formatting tags, it will now be rendered as HTML.

TAdvRichEditor

- New: event OnMergeField added

The event OnMergeField is triggered during a merge. It is triggered for each merge field in the richeditor document and allows to dynamically replace the merge field with the merge field value. The signature of the OnMergeField event is:

```
MergeField(Sender: TObject; MergeName: string; var MergeValue: string; var
DoMerge: Boolean);
```

The MergeName parameter contains the name of the merge field. The var parameter MergeValue can be used to set the merged value and with the DoMerge parameter, it can be controlled whether the merge will be performed or not.

TDBAdvRichEditor

- New: dynamic merging of DB fields

When a DBAdvRichEditor.MergeSource is set and merge fields are added to the TDBAdvRichEditor document, as soon as the MergeSource dataset is activated, the merge will be executed. When record values change in the dataset or the dataset is navigated, the merge field values will be live updated.

TAdvEdit, TAdvMaskEdit, TAdvSpinEdit

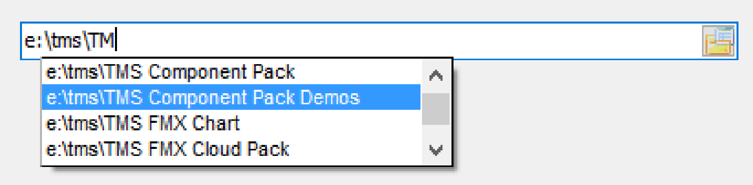
- New: Events OnAfterClipboardCut, OnAfterClipboardCopy, OnAfterClipboardPaste added

New events notify after a clipboard operation has happened and can be used to work on the content of the edit control after the clipboard operation.

TAdvFileNameEdit

- New: property AutoFileLookup added

When AutoFileLookup is set to true, the lookup dropdown is automatically filled with matching filenames retrieved from the local file system.



e:\tms\TM

e:\tms\TMS Component Pack
e:\tms\TMS Component Pack Demos
e:\tms\TMS FMX Chart
e:\tms\TMS FMX Cloud Pack

TAdvDirectoryEdit

- New: property AutoDirectoryLookup added

When AutoDirectoryLookup is set to true, the lookup dropdown is automatically filled with matching directory names retrieved from the local file system.

TInspectorBar / TFormControlInspectorEditLink

- New: TFormControlInspectorEditLink component

This new component allows to use any TWinControl based edit control that is placed on the form as inplace editor for the TInspectorBar.

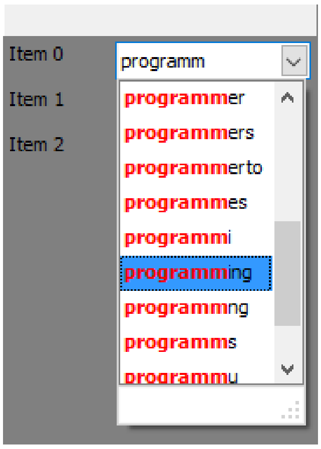
To start using TFormControlInspectorEditLink, drop an instance on the form and also an instance of the edit control you want to use as inplace editor. Assign this control to TFormControlInspectorEditLink.Control. Then hook up the EditLink to TInspectorItem.EditLink and set TInspectorItem.PropertyType to ptCustom. Implement minimum the TFormControlInspectorEditLink.OnGetEditorValue / TFormControlInspectorEditLink.OnSetEditorValue events to get & set the value of the TInspectorItem as text from the control value.

Example:

To use a TAdvSearchEdit control as inplace editor for a TInspectorBar item, that performs filtering / lookup while typing in a dictionary, following code can be used applied on a default TInspectorBar:

```
procedure TForm1.FormCreate(Sender: TObject);
var
  sl: TStringList;
  i: integer;
begin
  AdvSearchEdit1.SearchButton.Visible := false;
  AdvSearchEdit1.CategoryButton.Visible := false;
  AdvSearchEdit1.DropDownHeader.Visible := false;

  // loading the dictionary file in the TAdvSearchEdit
  sl := TStringList.Create;
  try
    sl.LoadFromFile('e:\tms\temp\dictionary.txt');
    AdvSearchEdit1.LoadStrings(sl);
  finally
    sl.Free;
  end;

  // linking up the TAdvSearchEdit as inplace editor
  FormControlInspectorEditLink1.Control := AdvSearchEdit1;
  // inplace editor control will still hanbdle return key
  FormControlInspectorEditLink1.WantKeyReturn := true;

  InspectorBar1.Panels[0].ItemHeight := 26;
  InspectorBar1.Panels[0].Style := psProperties;

  for i := 0 to InspectorBar1.Panels[0].Items.Count - 1 do
  begin
    InspectorBar1.Panels[0].Items[i].EditLink :=
FormControlInspectorEditLink1;
    InspectorBar1.Panels[0].Items[i].PropertyType := ptCustom;
  end;
end;

procedure TForm1.FormControlInspectorEditLink1GetEditorValue(Sender:
TObject;
  Item: TInspectorItem; var AValue: string);
begin
  AValue := AdvSearchEdit1.Text;
end;

procedure TForm1.FormControlInspectorEditLink1SetEditorValue(Sender:
TObject;
  Item: TInspectorItem; AValue: string);
begin
  AdvSearchEdit1.Text := AValue;
end;
```

The result looks like:



TAdvPolyList

- New: property TButtonBarElement.Down: Boolean property added

Gets or sets the down state of a button in a TButtonBarElement.

- New: OnAfterItemReorder event added

Event is triggered after a reordering of the items happened with the mouse. The signature of the event is:

```
OnItemChangeValue(Sender: TObject; Index: Integer; var Value: Double);
```

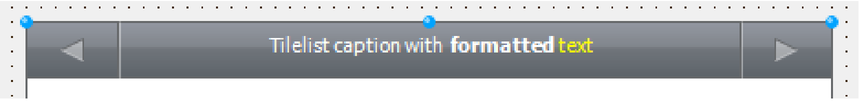
It returns the index of the TCapacityItem that is being changed and its new value.

TAdvSmoothTileList

- New: capability to show HTML formatted text in header and footer added

When TAdvSmoothTileList.Header.ShowPages = false, a caption can be shown in the header and this can now contain HTML formatted text:

```
begin
  AdvSmoothTileList1.Header.Caption := 'Tilelist caption with
<b>formatted</b> <font color="clYellow">text</font>';
end;
```



TDBAdvSmoothTimeLine

- New: IndicatorHint databinding capability added

Under TDBAdvSmoothTimeLine.DataBinding, the new property IndicatorHint allows to set the DB field that needs to be used to set the indicator hint value.

TAdvSmoothSlideShow

- New: Capability added to load slideshow images also from a stream

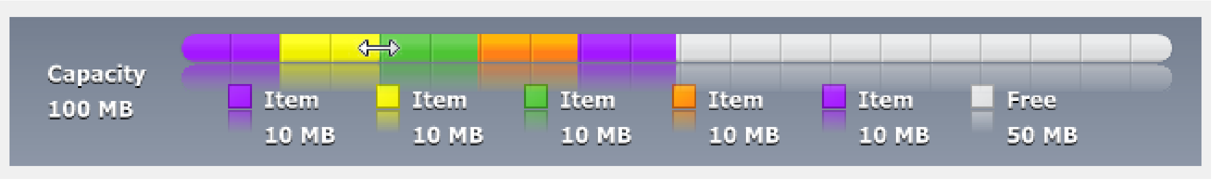
```
AdvSmoothSlideShow1.AddItemFromStream(AStream, ikImageText, 'Description', 'Caption', 'Hint');
```

TAdvSmoothCapacityBar

- New: TCapacityItem.AllowChange property added

- New: OnItemChangeValue event added

When TCapacityItem.AllowChange = true, it is possible to change the value of the capacity item by dragging the item by its right-side border. When the value is being changed, the OnItemChangeValue event is triggered.



TAdvSmoothExpanderPanel / TAdvSmoothExpanderButton

- New: TAdvSmoothExpanderPanel.Columns: integer, TAdvSmoothExpanderPanel.Row: integer added

- New: TAdvSmoothExpanderPanel.AlternativeLayout: Boolean property added

- New: TAdvSmoothExpanderButton.ColumnSpan, TAdvSmoothExpanderButton.RowSpan properties added

On a regular TAdvSmoothExpanderPanel, the buttons are laid out proportionally on the surface of the panel and all buttons have the same size (AdvSmoothExpanderPanel.ButtonWidth, AdvSmoothExpanderPanel.ButtonHeight). When the new AlternativeLayout: Boolean property is set to true, it is possible to add buttons to the layout that span multiple rows or multiple columns or both. This way, buttons with different sizes can be added to the TAdvSmoothExpanderPanel.

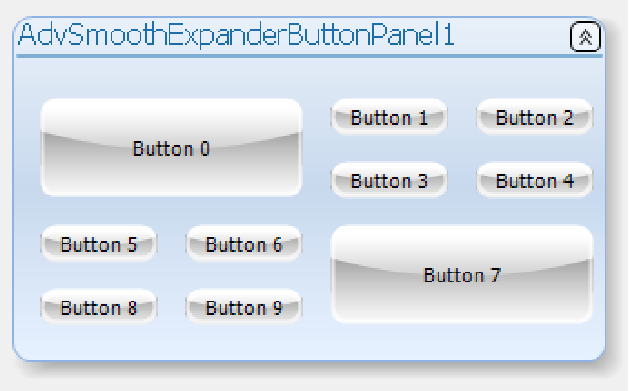
To use the new feature, make sure to set TAdvSmoothExpanderPanel.AlternativeLayout = true and set both TAdvSmoothExpanderPanel.Columns, TAdvSmoothExpanderPanel.Rows to a value different from -1. Set the TAdvSmoothExpanderButton.ColumnSpan, TAdvSmoothExpanderButton.RowSpan to values bigger than 1 where larger buttons are desired.

Example:

A default TAdvSmoothExpanderPanel panel initialized with:

```
var
  i: integer;
begin
  AdvSmoothExpanderButtonPanel1.AlternativeLayout := true;
  AdvSmoothExpanderButtonPanel1.Columns := 4;
  AdvSmoothExpanderButtonPanel1.Rows := 4;
  AdvSmoothExpanderButtonPanel1.Buttons.Clear;

  for i := 0 to 9 do
    AdvSmoothExpanderButtonPanel1.Buttons.Add.Caption := 'Button '+
inttostr(i);

  AdvSmoothExpanderButtonPanel1.Buttons[0].ColumnSpan := 2;
  AdvSmoothExpanderButtonPanel1.Buttons[0].RowSpan := 2;

  AdvSmoothExpanderButtonPanel1.Buttons[7].ColumnSpan := 2;
  AdvSmoothExpanderButtonPanel1.Buttons[7].RowSpan := 2;
end;
```

results in:



TMS Component Pack v8.3.4.0 - Sep 23, 2016

TAdvDateTimePicker

- New: MinTime, MaxTime properties added

Now it is possible to also specify a minimum time and maximum time value for allowed time to enter in the TAdvDateTimePicker.

TAdvGridFilterPanel

- New: OnAppliedFiter, OnRemovedFilter, OnRestoredFilter events added

Extra events are added that are triggered after each of the filter, clear filter, load filter is done in the grid filter panel to enable to do appropriate actions after the filtering is updated.

T(DB)Planner

- New: Planner.Items.SelectAll method added

When Planner.MultiSelect = true, all items can now be set in selected state programmatically with the method Planner.Items.SelectAll.

TAdvCardList

- New: Swipe property added

When AdvCardList.Swipe is set to true, it is now possible to scroll through the card list control with a click and drag operation.

TAdvTaskDialog

- New: FooterTextColor property added

It is possible now to specify the text color for the footer text in the task dialog. The property FooterTextColor is used when TAdvTaskDialog.NonNativeDialog is set to nndAlways.

TMS Component Pack v8.3.3.0 - Sep 5, 2016

TAdvStringGrid

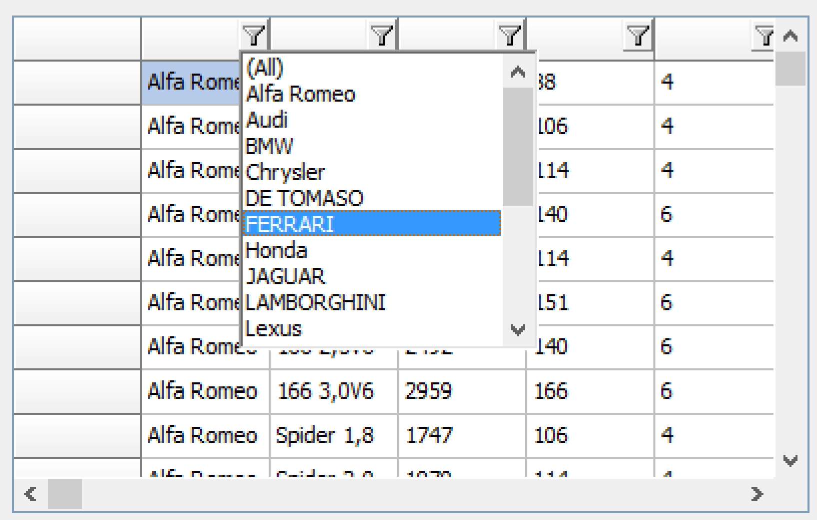
- New: ShowFilterDropDown method added

Now is it possible to programmatically show the filter dropdown for a grid column. When auto filtering is enabled, the filter is shown with: grid.ShowFilterDropDown(Column).

This way it is possible to hook-up the filter dropdown also to a hotkey for example instead of having it shown only when the mouse clicks the filter icon.

This code snippet how the Ctrl-F5 hotkey can show the filter dropdown for example:

```
procedure TForm1.AdvStringGrid1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if (ssCtrl in Shift) and (Key = VK_F5) then
    AdvStringGrid1.ShowFilterDropDown(AdvStringGrid1.Col);
end;
```



	Alfa Rome			38	4
	Alfa Rome			106	4
	Alfa Rome			114	4
	Alfa Rome			140	6
	Alfa Rome			114	4
	Alfa Rome			151	6
	Alfa Rome			140	6
	Alfa Romeo	166 3,0V6	2959	166	6
	Alfa Romeo	Spider 1,8	1747	106	4

TMS Component Pack v8.3.2.0 - Sep 1, 2016

Delphi 10.1 Berlin Starter Edition support

We have now added support for Delphi 10.1 Berlin Starter edition. As Embarcadero disables the command-line compiling in a starter edition, our automatic installer invoking this command-line compiler cannot compile the component sources for you upon installation. Therefore, we included a project group file especially for the Delphi 10.1 Berlin Starter Edition that allows you to install the components in the IDE. After running the installer and have it extract all sources, open the group tmsdXE10startergroup.groupproj from the IDE and compile & install each package in this group from the project manager in the IDE.

TAdvStringGrid

- New: OnHoverButtonsShow event

This new event enables a much higher degree of configurability and flexibility for using hover buttons on TAdvStringGrid. Whereas previously, it was only possible to have a fixed set of hover buttons appearing for every row, now it is possible to control hover buttons per cell.

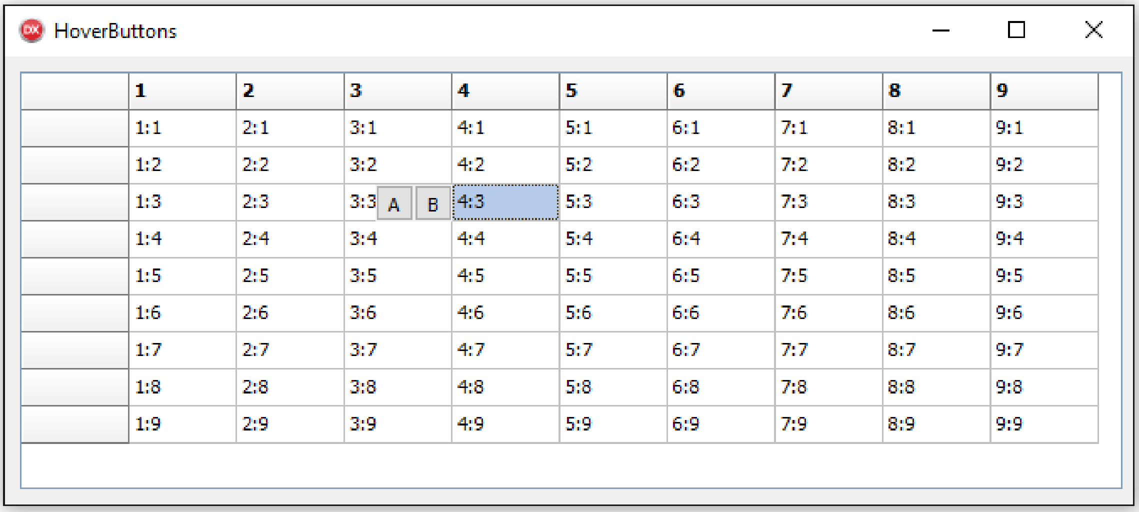
Example showing different hover buttons for columns 4 and 6:

To see this in action, enable HoverButtons and implement the OnHoverButtonsRow event that is triggered when the mouse hovers a cell. Here it is possible to control whether hover buttons are shown with the Allow event parameter and from this event, the grid.HoverButtons property can be used to customize hover buttons per cell.

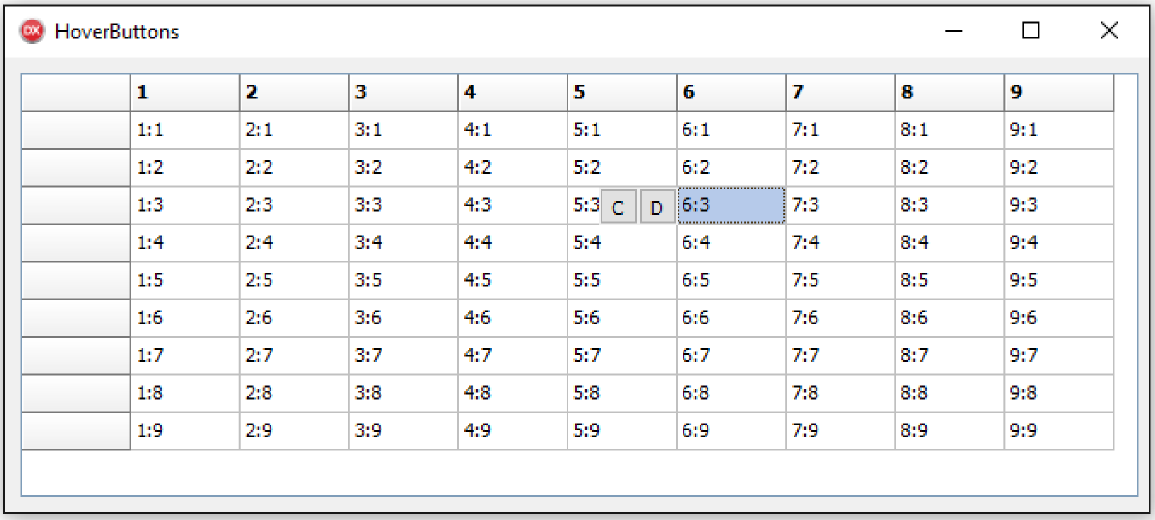
The code to show different hover buttons only for column 4 and 6 is therefore:

```
procedure TForm1.AdvStringGrid1HoverButtonsShow(Sender: TObject; X, Y:
Integer;
  var Allow: Boolean);
var
  c,r: integer;
begin
  AdvStringGrid1.MouseToCell(x,y,c,r);

 if c = 4 then
  begin
    // configure hover buttons here for column 4
    AdvStringGrid1.HoverButtons.Buttons[0].Caption := 'A';
    AdvStringGrid1.HoverButtons.Buttons[1].Caption := 'B';
    AdvStringGrid1.HoverButtons.Column := 4;
  end
  else
  if c = 6 then
  begin
    // configure hover buttons here for column 6
    AdvStringGrid1.HoverButtons.Buttons[0].Caption := 'C';
    AdvStringGrid1.HoverButtons.Buttons[1].Caption := 'D';
    AdvStringGrid1.HoverButtons.Column := 6;
  end
  else
    Allow := false;

end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  AdvStringGrid1.HoverButtons.Enabled := true;
  AdvStringGrid1.HoverButtons.Buttons.Clear;
  AdvStringGrid1.HoverButtons.Buttons.Add.Caption := '0';
  AdvStringGrid1.HoverButtons.Buttons.Add.Caption := '1';
  AdvStringGrid1.AutoNumberRow(0);
  AdvStringGrid1.LinearFill(false);
end;
```

Result:



	1	2	3	4	5	6	7	8	9
	1:1	2:1	3:1	4:1	5:1	6:1	7:1	8:1	9:1
	1:2	2:2	3:2	4:2	5:2	6:2	7:2	8:2	9:2
	1:3	2:3	3:3	4:3	5:3	6:3	7:3	8:3	9:3
	1:4	2:4	3:4	4:4	5:4	6:4	7:4	8:4	9:4
	1:5	2:5	3:5	4:5	5:5	6:5	7:5	8:5	9:5
	1:6	2:6	3:6	4:6	5:6	6:6	7:6	8:6	9:6
	1:7	2:7	3:7	4:7	5:7	6:7	7:7	8:7	9:7
	1:8	2:8	3:8	4:8	5:8	6:8	7:8	8:8	9:8
	1:9	2:9	3:9	4:9	5:9	6:9	7:9	8:9	9:9



	1	2	3	4	5	6	7	8	9
	1:1	2:1	3:1	4:1	5:1	6:1	7:1	8:1	9:1
	1:2	2:2	3:2	4:2	5:2	6:2	7:2	8:2	9:2
	1:3	2:3	3:3	4:3	5:3	6:3	7:3	8:3	9:3
	1:4	2:4	3:4	4:4	5:4	6:4	7:4	8:4	9:4
	1:5	2:5	3:5	4:5	5:5	6:5	7:5	8:5	9:5
	1:6	2:6	3:6	4:6	5:6	6:6	7:6	8:6	9:6
	1:7	2:7	3:7	4:7	5:7	6:7	7:7	8:7	9:7
	1:8	2:8	3:8	4:8	5:8	6:8	7:8	8:8	9:8
	1:9	2:9	3:9	4:9	5:9	6:9	7:9	8:9	9:9

TAdvGridColumnPicker

- New: event OnAllowColumnDrop event added

With this new event, it is possible to control that at specific column positions, no column can be dropped.

Example:

With a TAdvGridColumnPicker connected to a grid and the code:

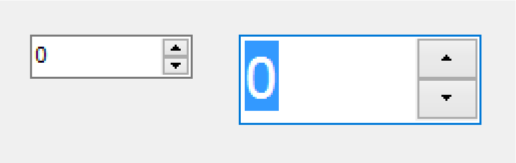
```
procedure TForm1.AdvGridColumnPicker1AllowColumnDrag(Sender: TObject;
  ACol: Integer; var Allow: Boolean);
begin
  Allow := ACol > 2;

end;

procedure TForm1.AdvGridColumnPicker1AllowColumnDrop(Sender: TObject;
  ACol: Integer; var Allow: Boolean);
begin
  Allow := ACol > 2;
end;
```

the first 2 normal columns cannot participate in the drag & drop to a column picker, effectively making just these 2 normal columns locked.

TAdvSpinEdit

- New: SpinButtonWidth property added

Now it is possible to also set the width of the spin up/down buttons, especially handy for using the control on touchscreen applications



TAdvOutlookList

- New: Events OnMouseLeave, OnMouseEnter exposed

TMS Component Pack v8.3.1.0 - Aug 18, 2016

TAdvResponsiveList

- New: Keyboard support added

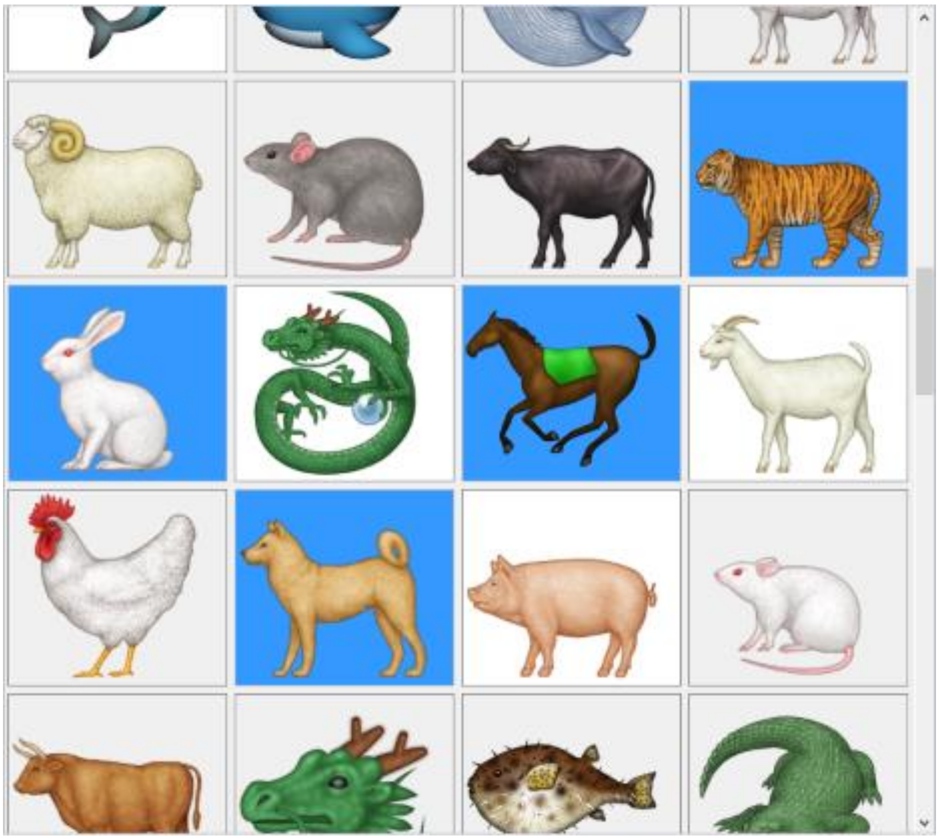
Now arrow keys, Home, End, Prior, Next keys can be used to select items.

- New: OnBeforeItemSelect, OnAfterItemSelect events added

New event triggered before an item is selected and allowing to cancel the selection of an item and an even after the item selection.

- New: Multiselect capability added

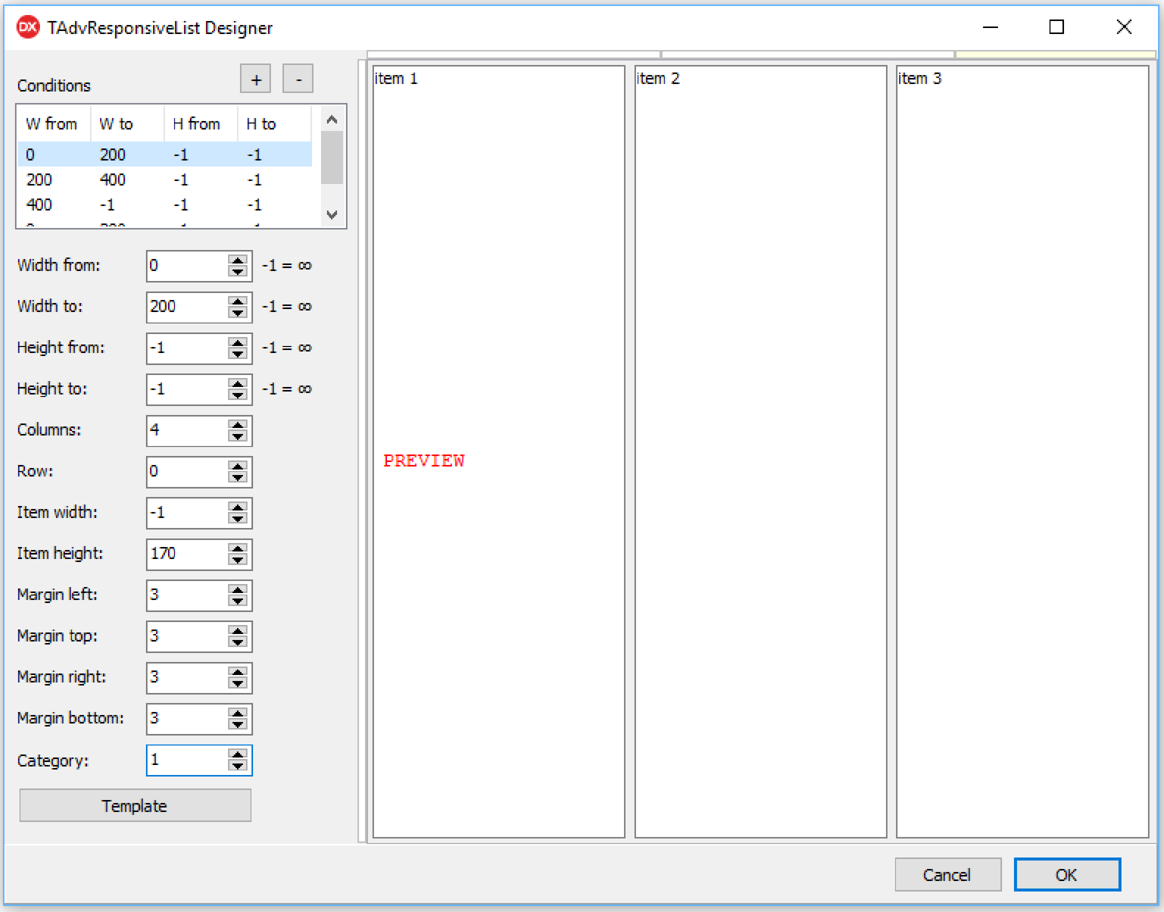
When MultiSelect = true, disjunct multi select of items is possible. Ctrl-Click can be used to add/remove items from the selection and Click, Shift-Click can be used to select a range of items.



The selected state can in this case be retrieved via AdvResponsiveList.Items[x].Selected: Boolean;

- New: ConditionCategory

Each condition can be assigned a category and the category to use can be set via the ConditionCategory property. This allows for example to set different conditions for different modes of operation of the TAdvResponsiveList.



W from	W to	H from	H to
0	200	-1	-1
200	400	-1	-1
400	-1	-1	-1

Width from: 0 -1 = ∞
Width to: 200 -1 = ∞
Height from: -1 -1 = ∞
Height to: -1 -1 = ∞
Columns: 4
Row: 0
Item width: -1
Item height: 170
Margin left: 3
Margin top: 3
Margin right: 3
Margin bottom: 3
Category: 1

- New: Touch scroll support

Now, touch and immediately scroll in the list by dragging is possible.

- New: OnBackgroundDraw() event

Event triggered before items are drawn and allows to customize the drawing of the background.

- New: Support for displaying controls in items

Via HTML tags, it is now possible to display and use controls CheckBox, RadioButton and Button in the item.

- New: OnItemControlClick() event

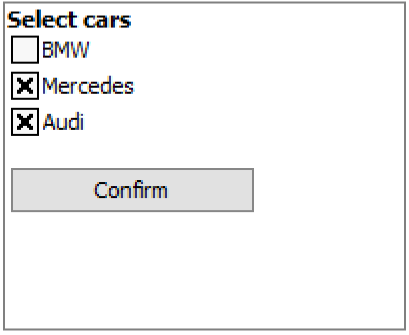
The event is triggered when a control in the HTML content of an item is clicked.

Example:

Here in the first item, 3 checkboxes and one button control is inserted in the item:

```
  AdvResponsiveList1.Items[0].Content := '<b>Select cars</b><br><control type="check"
id="a">BMW<br><control type="check" id="b">Mercedes<br><control type="check"
id="c">Audi<br><br><control type="button" id"btn" value="Confirm" width="125">';
```

resulting in:



Select cars
☐ BMW
☒ Mercedes
☒ Audi

TDBPlanner

- New: Upgraded ResourceIndex in ResourceMap to Int64 for allowing a larger range of resource indexes

To allow to also directly map on 64bit DB key fields, the ResourceIndex property is now of the type int64.

TInspectorBar

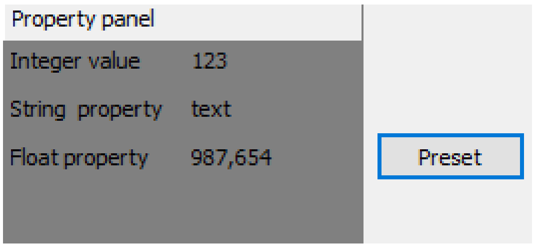
- New: InspectorItem.FloatValue property added

In addition to InspectorItem.IntValue that can be used to get & set integer property values, the InspectorItem.FloatValue allows to do the same for items that have PropertyType = ptFloat.

Example:

In this inspector panel, the property is updated as float with:

```
  InspectorBar1.Panels[0].Items[2].FloatValue := 987.654;
```



Property panel

Integer value	123
String property	text
Float property	987,654

TAdvListView

- New: ssAnsiAlphaCase , ssAnsiAlphaNoCase sort modes added

Allow to sort with taking special characters in account like accented characters with and without case sensitivity.

TAdvEdit

- New: AllowShares property added in TAdvDirectoryEdit

When AllowShares = true, the directory picker will also show mapped network drives.

- New: Public Value: variant property added to get & set value as variant

Allows to get and set the value of TAdvEdit via a variant.

- New: DisabledTextColor property added in TAdvEdit

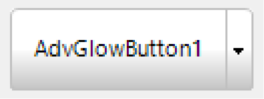
Sets the color of the edit control text when it is disabled. The default Windows disabled color is used when DisabledTextColor = clNone

TAdvGlowButton

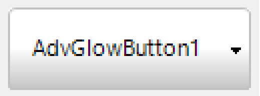
- New: Public class property DrawSplitLineAlways: boolean added

When true, the split line between the button and its dropdown part is always drawn, otherwise it is only drawn in hot & down state. This is a class property, so affects all TAdvGlowButtons in the application.

With DrawSplitLineAlways = true



With DrawSplitLineAlways = false



- New: UseAppearance property to force a glow button to use its own style regardless of toolbar style

When UseAppearance is set to true, the TAdvGlowButton does not use the appearance that is forced by the common appearance of a toolbar styler but it keeps using its own appearance settings. This way, a button can still have a different color than the Office theme color chosen for a toolbar.

TAdvStringGrid

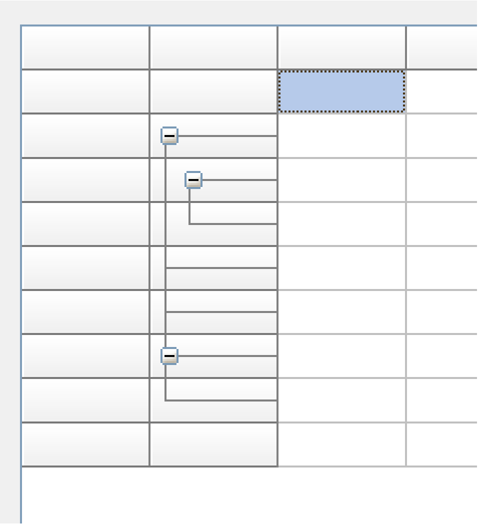
- New: CellNode.NodeColumn property

With this property it is now possible to control in what column of the grid nodes will be displayed. Formerly, nodes were always displayed in column 0. To have the nodes in a different column, simply set grid.CellNode.NodeColumn := NewColumn. Note that the node column can only be changed when the grid has no nodes yet, i.e. add the nodes after setting grid.CellNode.NodeColumn.

Example:

This code snippet adds nodes in the 2nd fixed column instead of the default first fixed column.

```
begin
  AdvStringGrid1.FixedCols := 2;
  AdvStringGrid1.CellNode.NodeColumn := 1;

  AdvStringGrid1.AddNode(2,5);
  AdvStringGrid1.AddNode(3,2);
  AdvStringGrid1.AddNode(7,2);
end;
```



- New: HTML rendering engine float attribute for paragraph & image

Both IMG and P tag can have a new attribute float. For an image, the float attribute can be "left" or "right". For a P tag, the float attribute can be "left". With the float attribute it is possible to have an image float left or right and have text rendered on multiple lines next to the image. For text it is possible to have both left and right aligned text on the same line for example.

Example:

<img src="picturename" float="left">This is multiline text that is displayed next to the image

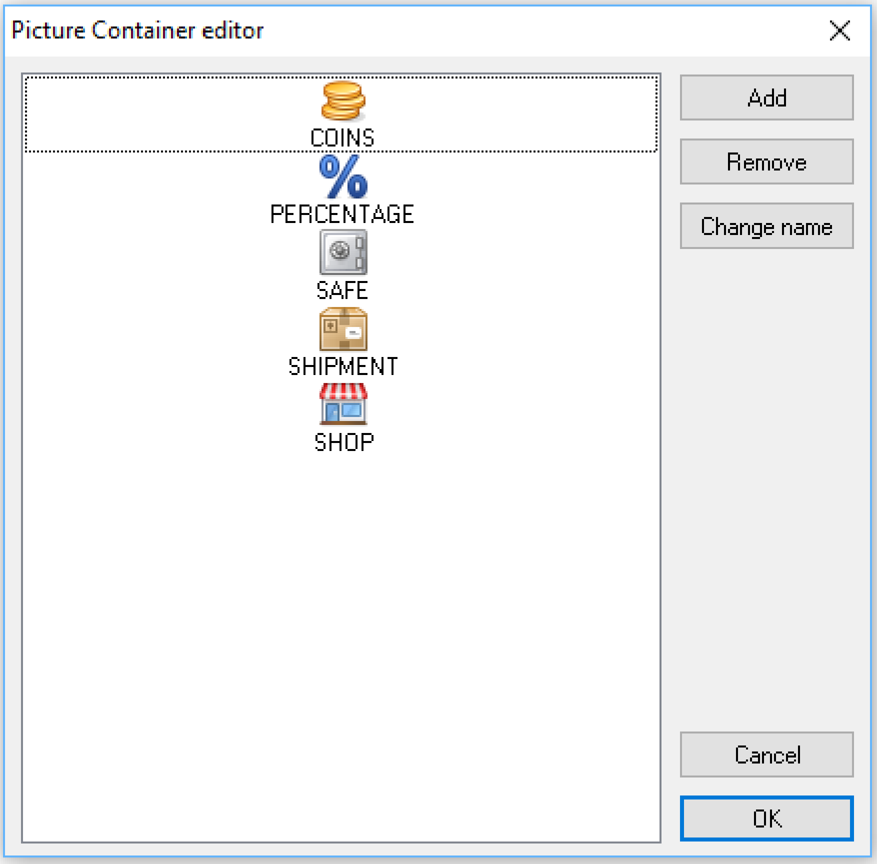
<p align="left" float="left">Product A</p><p align="right">123.456</p>

- New: AddDataPicture, RemoveDataPicture, HasDataPicture methods added

It is now possible to attach a picturecontainer to the grid and specify the image to display in a cell by simply setting the picture name it has in the picturecontainer in the cell.

Example:

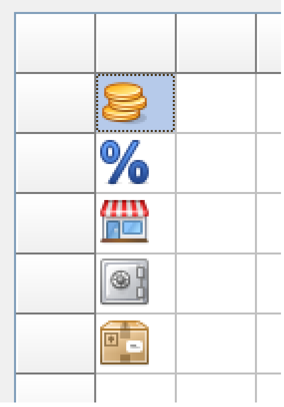
Take a PictureContainer initialized as:



And initialize the grid with the code:

```
var
  i: integer;
begin
  for i := 1 to AdvStringGrid1.RowCount - 1 do
    AdvStringGrid1.AddDataPicture(1,i,",haLeft,vaTop);

  AdvStringGrid1.Cells[1,1] := 'COINS';
  AdvStringGrid1.Cells[1,2] := 'PERCENTAGE';
  AdvStringGrid1.Cells[1,3] := 'SHOP';
  AdvStringGrid1.Cells[1,4] := 'SAFE';
  AdvStringGrid1.Cells[1,5] := 'SHIPMENT';
end;
```

The result is:



- New: Find function with search for strings containing special chars like &, <,>,=,! when using dbl quotes

Using the Find function, it is now also possible to search on text matches where text has special characters such as <, >, =, ! that were previously used as search specifiers when fnMatchRegular style was used.

Example:

When a grid cell contains the text: M&G, it is now possible to search for a matching cell via a specifier “M&*”. Note that the double-quotes must be used in case these special characters are part of the search string.

TAdvSmoothTimeLine

- New: EndTimeMode property to allow sections to end at Now or at the fixed EndTime property

Allows to specify whether a section will end at the current time or will use the value as end time set by the EndTime property.

TAdvMenus

- New: ActionImages property added to select type of images for TActionManager associated menu items

Via the TAdvMenu.ActionImages and TAdvPopupMenu.ActionImages property it can now be specified to use the normal ActionManager images or the large images. Use aiNormal for the normal size images and aiLarge for the large images imagelist.

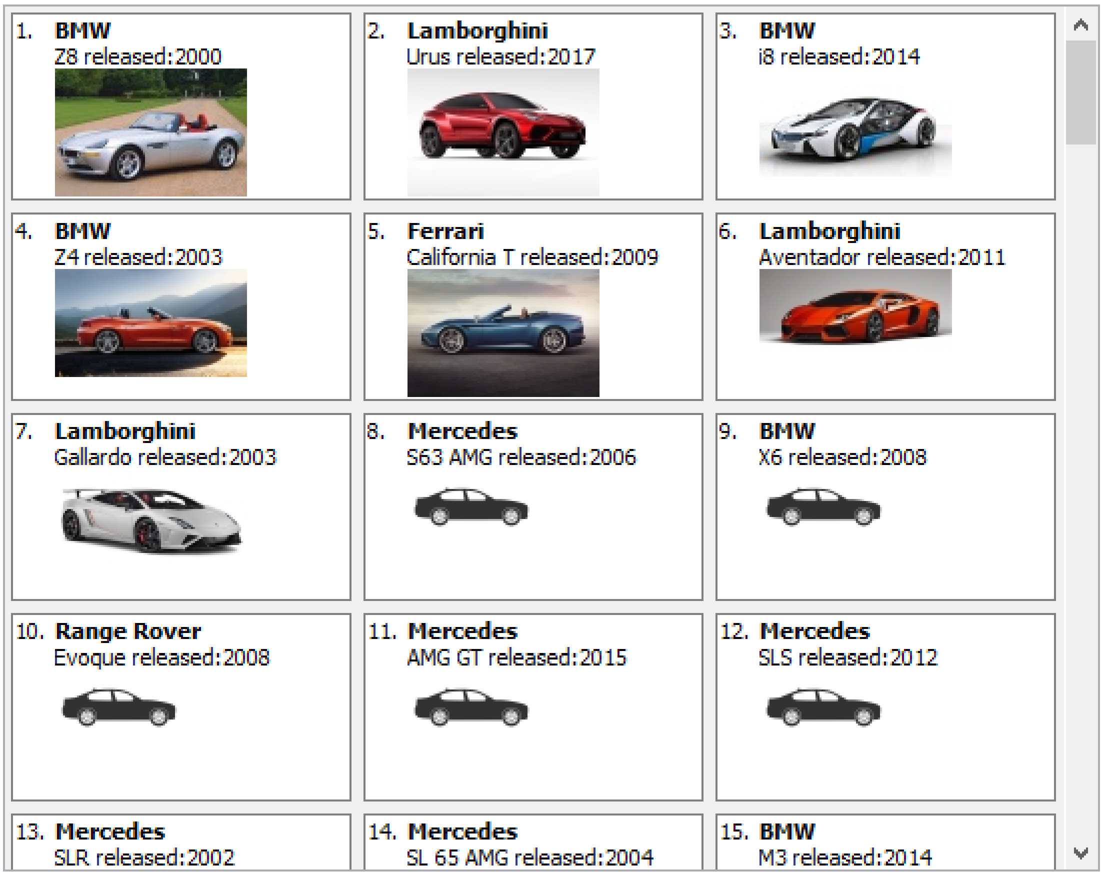
TWebPictureContainer

- New: LoadPicture added to control load mechanism

With the new LoadPicture property, it can be controlled whether the TWebPictureContainer background thread will try to load all pictures in its container or will only load pictures needed. A picture is considered needed when the control using the PictureContainer requests the image. Normally, a control will only request PictureContainer pictures when it needs it to display. The default is lpAlways, meaning that as soon as items are added with an URL to the PictureContainer, the background thread will kick in action to retrieve the picture. When LoadPicture = lpOnRequest, only when a control requests a picture, the background thread will kick in action to retrieve it. This also means that for the first request, the DefaultPicture will be returned to the control and when the picture is retrieved, the PictureContainer will trigger a display update in the UI control to have it display the final retrieved picture.

Example:

This TAdvResponsive list control is filled with pictures via URL references. The black & white picture is the default picture. This screenshot was captured as the background thread was resolving the URL’s one after one:



1. **BMW**
Z8 released:2000

2. **Lamborghini**
Urus released:2017

3. **BMW**
i8 released:2014

4. **BMW**
Z4 released:2003

5. **Ferrari**
California T released:2009

6. **Lamborghini**
Aventador released:2011

7. **Lamborghini**
Gallardo released:2003

8. **Mercedes**
S63 AMG released:2006

9. **BMW**
X6 released:2008

10. **Range Rover**
Evoque released:2008

11. **Mercedes**
AMG GT released:2015

12. **Mercedes**
SLS released:2012

13. **Mercedes**
SLR released:2002

14. **Mercedes**
SL 65 AMG released:2004

15. **BMW**
M3 released:2014

- New: Capability to load from file added

When the prefix http:// is not used for the URL of the PictureItem, now TWebPictureContainer will try to load the picture from the local file system.

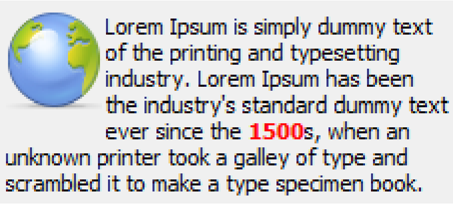
HTML controls

- New: HTML rendering engine float attribute for paragraph & image

Both IMG and P tag can have a new attribute float. For an image, the float attribute can be “left” or “right”. For a P tag, the float attribute can be “left”. With the float attribute it is possible to have an image float left or right and have text rendered on multiple lines next to the image. For text it is possible to have both left and right aligned text on the same line for example.

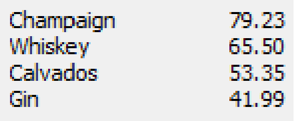
Example:

```
<img src="globe" float="left">Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's standard dummy text ever since the <B><FONT
color="#FF0000">1500</FONT></B>s, when an unknown printer took a galley of type and scrambled
it to make a type specimen book.
```



Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the **1500**s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

```
<p align=”left” float=”left”>Product A</p><p align=”right”>123.456</p>

<p align="left" float="left">Champaign</p><p align="right">79.23</p>
<p align="left" float="left">Whiskey</p><p align="right">65.50</p>
<p align="left" float="left">Calvados</p><p align="right">53.35</p>
<p align="left" float="left">Gin</p><p align="right">41.99</p>
```



Champaign	79.23
Whiskey	65.50
Calvados	53.35
Gin	41.99