# TMS TAdvStringGrid what's new guide

Productivity feature packed grid for Delphi & C++Builder

# Introduction

TMS TAdvStringGrid v6.1 is the latest version of the TMS productivity feature packed grid for CodeGear Delphi & C++Builder. While many new capabilities and improvements are introduced, it is designed to be fully backwards compatible with TAdvStringGrid v3.0 or later. Upgrading applications to the latest version of TAdvStringGrid should as such be seamless.

# Availability

TMS TAdvStringGrid is available as a VCL component set for Win32 application development.  TMS TAdvStringGrid is available for Embarcadero™ Delphi 5,6, 7,2005,2006,2007,2009,2010,XE,XE2 & Embarcadero™ C++Builder 5,6,2006,2007,2009,2010,XE,XE2.

# What's new & improved in version v6.1

### New : CursorMoveRows navigation mode

When the property grid.Navigation.CursorMoveRows is set to true, pressing Ctrl and arrow up or down moves the selected row or rows up or down. When range selection is enabled and multiple rows are selected, pressing Ctrl-Down, Ctrl-Up will move the entire selected block of rows one row up or down. When grid.Navigation.CursorMoveRows is false, pressing Ctrl-Up or Ctrl-Down has no effect.

### New : OnFilterCheck event

When the option is chosen to have a filter dropdown with checkboxes, i.e. when grid.FilterDropDownCheck = true, the event OnFilterCheck is triggered when a checkbox in the filter dropdown is clicked.

### New : FilterDropDownColumns property

Selects whether the filter dropdown appears for all columns or only for normal columns (excluding fixed columns). Values are fdAll or fdNormal.

### New : edMaskEditBtn inplace editor type

This new inplace editor consists of an edit control with a mask and a button attached. The edit control's mask is set via the event OnGetEditMask. The button will trigger the event OnEllipsClick when clicked.

### New : adLeftRightInRow, adTopBottomInCol navigation modes

The property to control the direction of moving to the next cell when grid.Navigation.AdvanceEnter = true and the Enter key is pressed is extended now with two extra options. Following options are now available:

adTopBottom: move from top row to bottom row and then move to next column.

adTopBottomInCol: move from top row to bottom row and then move to top row again in same column.

adLeftRight: move from left to right and then move to next row.

adLeftRightInRow: move from left to right and then move to back to first cell in same row.

### New : MoveRows(FromIndex, ToIndex, RCount: integer) method added

Programmatically moves RCount rows starting from index FromIndex to ToIndex. This complements the MoveRow() method that just moves one row.

**New : Basic support for XE2 VCL Styles**

The grid will now use the VCL Style definitions for cell background, text color, fixed cell backgrounds and text color and automatically synchronize with the selected VCL style. When the VCL style is the classic Windows style, the grid appearance is unchanged and not affected.

# What's new & improved in version v6.0

**New : Metro style**

The grid can now be put in a minimalistic Metro design language style with the method grid.SetColorTones(). This function will be integrated in other TMS Components as well and eventually in the TAdvFormStyler/TAdvAppStyler to change applications fully to Metro style with one setting.

**New : Tristate checkbox support added**

New overrides of the methods to add checkboxes in the grid are provided with the State: TCheckBoxState parameter to have tri-state checkboxes in the grid.

**New : TAdvHeaderList / TAdvHeaderListPopup column chooser components**

Two new components make it easy to perform drag & drop of columns between a grid and a component that holds the list of columns. A new paragraph in the developers guide is devoted to how this can be used.

**New : OnEditChange support added for fixed cell editing**

Fixed cell editing has been extended with support for regular edit control, edit combobox and list combobox and will trigger the OnEditChange now during editing.

**New : Added option to control export of images with TAdvGridExcelIO**

A new setting allows to decide to export images to an XLS file or not with TAdvGridExcelIO.

**New : OnGetCellCursor event added**

The event is triggered for each cell and allows to specify a cell specific cursor.

**New : FilterNormalCellsOnly property added to skip filter on node/summary rows**

When FilterNormalCellsOnly is set to true, filtering applies only to normal cells and not to cells in rows that have a node or summary rows of a group. That allows to perform filtering on grouped grids.

**New : Group checkboxes**

When grid.Grouping.AutoCheckGroup = true, a checkbox on a group header row will now perform a check/uncheck of all checkboxes on rows in the group. Equally, the group checkbox will reflect the state of checkboxes on rows within the group.

**New : edNumericUnitEditBtn, edFloatUnitEditBtn inplace editor types**

Two new editor types are provided to allow inplace unit edit control with accepting numbers of floating point values only.

**New : Groups grand total automatically displayed in floating footer line**

When group calculations are performed and the floating footer is visible, the groups grand total or calculation result is now displayed automatically in the floating footer.

### New : Public properties SaveFixedCols, SaveFixedRows added

When SaveFixedCols / SaveFixedRows more fine-grained control is now possible to determine in what fixed cells data from a file is imported or from what fixed cells data is exported. Set the existing grid.SaveFixedCells = false and use SaveFixedCols / SaveFixedRows

### New : Grid.ControlLook.DropDownCount property added

With this property, it can be set how many rows will be displayed in the dropdown of inplace combobox editors.

### New : FilterDropDownCheck to have multiselect dropdown filter

When true, the automatic filter dropdown is a checklist. With this checklist, it is possible to perform an OR-type filter operation of the conditions in the checklist dropdown that is displayed on the column header.

### New : Combo edit & combo list fixed cell editor

With the properties grid.MouseActions.FixedColsEditor , FixedRowsEditor, it can now be specified to use a normal edit control, editable combobox or list combobox for fixed cell editing.

### New : Inplace editor alignment will automatically match display alignment

Now, the inplace editor alignment is by default set identical to the alignment of the displayed text in the cell

### New : DISTINCT, STDDEV column & group calculation function added

Two new column & group calculation methods are added. DISTINCT calculates the number of unique values in cells in a column or group column while STDDEV calculates the standard deviation of values. The new functions are grid.ColumnDistinct, grid.ColumnStdDev and grid.GroupDistinct, grid.GroupStdDev.

### New : Cell control to expand/collaps rows

With the new function AddExpand, a small expand/collaps button can be added in the top-right corner of a cell. Clicking this button will toggle the row height between default row height and expanded row height. The expanded row height is a parameter of the AddExpand() method.

### New : Method grid.EditCell(Col,Row) added

With this new function, the grid inplace editor can be easily started with one call in cell Col,Row.

### New : grid.SortSettings.ToggleDirection method added

With this new method, the sort direction can be easily toggled before calling grid.QSort. This will automatically take care that the sort order changes alternatingly for each sort.

### New : Event OnComboDropDown added

This new event is triggered on dropdown of the inplace editable combobox (edComboEdit) and the inplace list combobox (edComboList)

### New : Support for XY scatter charts added in TAdvChartLink

TAdvChartLink has been extended to provide automatic linking of data in TAdvStringGrid with an X,Y scatter chart type in the TMS TAdvChartView or TAdvGDIPChartView.

### New : OnColumnCalc event added

This new event is called during custom calculation to differentiate from the OnGroupCalc event that is triggered duriung custum column calculations in a grouped grid.

### New : DragDropSettings.OleColumnsOnly property added

When DragDropSettings.OleColumnsOnly is set to true, an OLE drag & drop operation can only be started from the column header cells (Fixed row) This is suitable for situations where only drag & drop of entire columns should be allowed and not of other normal cells.

### New : DragDropSettings.OleColumnReorder property added

When this new setting is trued, OLE drag & drop can also perform column reordering. Use this in combination with DragDropSettings.OleColumnsOnly = true.

### Improved : Flat node appearance

Flat nodes are now more correct displayed with the inner + - sign perfectly centered.

### Improved : Behavior with cell scroll

Cell scrollbar size calculation and behavior is now resembling better real Windows control scroll handling

### Improved : TAdvGridImport selection between delimited & fixed column with files

The preview of data in the TAdvGridImportDialog is now always autosized and selection of delimiters has been enhanced, specifically when custom delimiters are used.

### Improved : Column OLE drag & drop

Column drag & drop is now handling the column drag image better.

### Improved : 64bit compatibility

Various improvements have been done for 64bit compatibility when TAdvStringGrid is used in 64bit apps.


# What's new & improved in version v5.0

### New : method grid.SelectCells()

This new method makes it easier to programmatically set the selected cells in the grid

### New : grid.MouseActions.PreciseNodeClick

When this property is set, only the node rectangle in a cell is sensitive for clicks to open or collaps the node. Otherwise, the entire cell can be clicked.

**New : ColumnSize.Row property**

New property makes it possible to specify what rows should be taken in account to perform automatic sizing of columns. This can be set to:

arAll: all rows are taken in account
arNormal: only normal rows are taken in account
arFixed: only fixed rows are taken in account

**New : DateTimePicker.WeekNumbers, DateTimePicker.ShowToday, DateTimePicker.ShowTodayCircle properties**

New properties allow to customize the dropdown calendar of the inplace datepicker control to show week numbers, to turn off the today date & selecte display.

**New : public property XYOffsetTopLeftOnly property**

By default, the grid.XYOffset property is used to set the offset for text from left/right and top/bottom in a cell. When XYOffsetTopLeftOnly is set to true, the offset is only taken in account from the left & top corner.

**New : URLShowInText property**

When true, any hyperlink of the type http://, http://, ftp://, nntp://, mailto: that occurs in the text of a cell is displayed in the URLColor and underlined and will act as a hyperlink when clicked. This means that if the cell content is "Visit http://www.tmssoftware.com for details", the hyperlink will be automatically underlined without any need to write HTML <A> tags. This is different from using just URLShow = true, where the entire cell text is treated as hyperlink or not.

**New : FilterDropDownAuto/FilterDropDownClear added**

The grid has a built-in capability to automatically create the filter consisting of an alphabetically sorted list of unique column values. To enable this feature, set grid.FilterDropDownAuto to true. In addition to the list of unique values, an entry is optionally added that can be used to clear the filter. The value of this entry is set by grid.FilterDropDownClear: string and defaults to '(All)'. When the auto filter capability is enabled, every column header automatically gets the filter dropdown icon. When the auto filter is only needed for some specific columns, the OnGetColumnFilter event can be used to remove the filter by simply clearing the prefilled filter list. This sample code shows how the auto filter can be set to only columns 1 and 2:
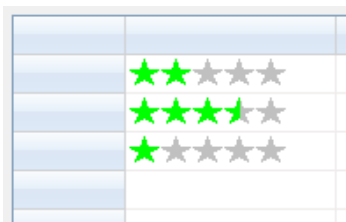
```
procedure TForm13.AdvStringGrid1GetColumnFilter(Sender: TObject;
 Column: Integer; Filter: TStrings);
begin
  if not (Column in [1,2])  then
    Filter.Clear;
end;
```

**New : Cell rating control**

As commonly seen in many web interfaces, the grid provides a specific graphic type to add a rating control to a cell. The rating control consists of a number of star shapes. The number of fully or half coloured stars represents the value. To add a rating control to a cell, call grid.AddRating(ACol,ARow,Scale: integer; FullColor,EmtpyColor: TColor). The parameters ACol,ARow set the cell where to add the rating control. The Scale parameter sets the number of stars to use in the rating control and the parameters FullColor,EmptyColor set the color for fully

coloured stars and non coloured stars. When the grid cell is not read-only, the value of the rating control can be change by clicking on the cell. The value can also be programmatically set and retrieved with grid.Floats[Col,Row]: double.  This example code adds 3 rating controls in the grid and presets its values:

```
begin
  AdvStringGrid1.AddRating(1,1,5,clLime,clSilver);
  AdvStringGrid1.Floats[1,1] := 2.0;
  AdvStringGrid1.AddRating(1,2,5,clLime,clSilver);
  AdvStringGrid1.Floats[1,2] := 3.6;
  AdvStringGrid1.AddRating(1,3,5,clLime,clSilver);
  AdvStringGrid1.Floats[1,3] := 1.1;
end;
```



### New : shapes in cells

Often it is desirable to have a little graphic representation and not having to resort to image resources for this makes it easy to have graphics in many colours and many shapes in a resource friendly way. The grid offers a range of such shapes that can be added via the method grid.AddShape(ACol,ARow: integer; Shape: TCellShape; FillColor,LineColor: TColor; hal: TCellHAlign; val: TCellVAlign);

The type TCellShape is defined in the unit AdvUtil and the supported shapes are:

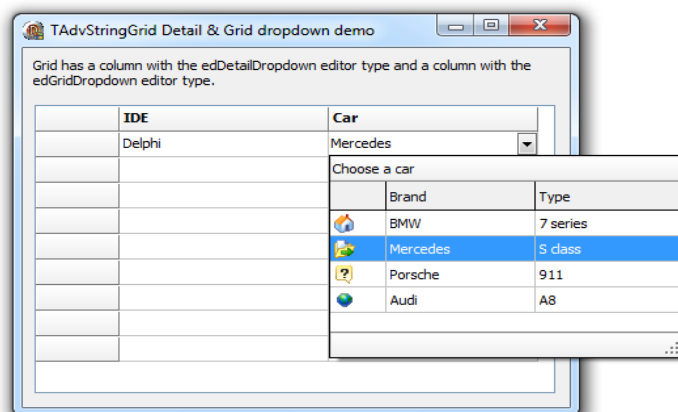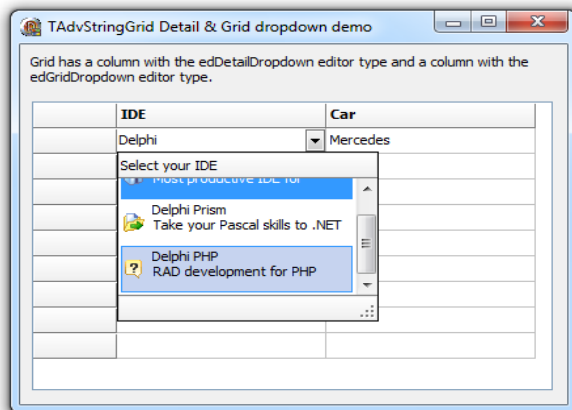| | |
|---|---|
| csRectangle | rectangular |
| csCircle | circle |
| csTriangleLeft | left oriented triangle |
| csTriangleRight | right oriented triangle |
| csTriangleUp | up oriented triangle |
| csTriangleDown | down oriented triangle |
| csDiamond | diamond shape |
| csLineVert | single vertical line |
| csLineHorz | single horizontal line |
| csRoundRect | rounded rectangle |
| csEllips | ellips |
| csSquare | square |
| csRoundSquare | rounded square |
| csStar | star |
| csArrowUp | arrow up |
| csArrowDown | arrow down |
| csArrowLeft | arrow left |
| csArrowRight | arrow right |
| csHalfStar | half star |

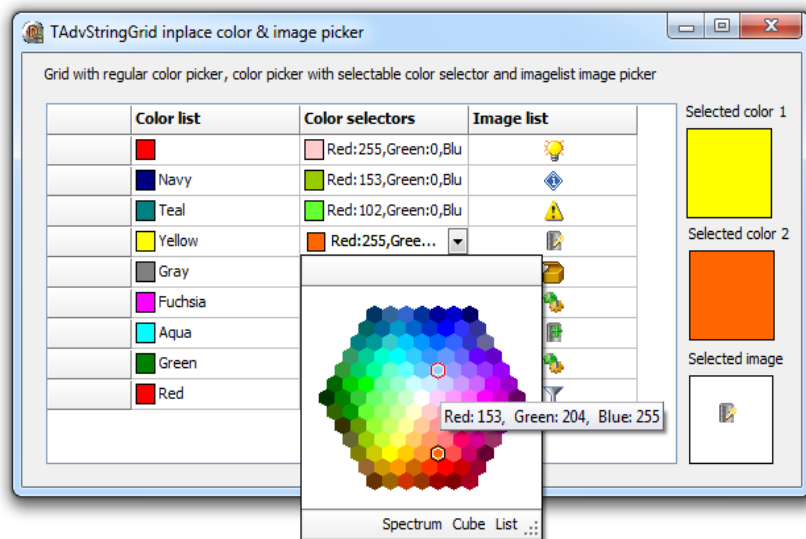To demonstrate this, following code adds 4 different shapes to the grid:

```
begin
  AdvStringGrid1.AddShape(1,1,csDiamond,clLime, clSilver,
haBeforeText,vaTop);
  AdvStringGrid1.AddShape(1,2,csCircle,clRed, clSilver,
haBeforeText,vaTop);
  AdvStringGrid1.AddShape(1,3,csStar,clYellow, clSilver,
haBeforeText,vaTop);
  AdvStringGrid1.AddShape(1,4,csArrowUp,clBlue, clSilver,
haBeforeText,vaTop);
end;
```



**New : detail,time,memo,colorpicker,imagepicker,trackbar,grid dropdown inplace editors**

A new range of feature-rich inplace editors is available now in TAdvStringGrid. The online samples:
http://www.tmssoftware.com/site/asg81.asp
http://www.tmssoftware.com/site/asg82.asp
explain how these new controls can be used.

**New : Fixed : column sizes can be preset in TAdvGridImportDialog**

Now it is possible to programmatically preset the column sizes before starting the TAdvGridImportDialog.

**New : property OriginalCells[Col,Row]: string**

This property enables to access cells irrespective of column moving.

**New : OnClipboardAfterPasteCell,OnClipboardAfterPasteWideCell events**

Event is triggered after the paste has been performed in a cell.

**New : MouseActions.DisjunctRowSelectNoCtrl property added**

When true and combined with DisjunctRowSelect = true, disjunct row selection is possible without keeping the Ctrl key down.

**New : Event OnCheckBoxChange**

The OnCheckBoxChange event is triggered whenever the state of a checkbox changes as a result from a click of the user on the grid. Note that the OnCheckBoxChange event will also be triggered for each checkbox changing its state by a click on a column header checkbox.

**New : Navigation.AutoAdvance new behaviour in combination with grid.MaxEditLength**

When Navigation.AutoAdvance is set to true, it will now also move to the next cell when MaxEditLength is reached

**New : Public property EditMask**

EditMask public property is now available to avoid implementing an extra event to set a mask

**New : public grid.ValidCharSet public property**

With this property it is possible to use a set of chars instead of a string of allowed characters

**Improved : IntelliZoom function**

Zoom in/out with mouse ctrl wheel change has better control over font sizing

### Improved : clipboard support for multiline cells

Copy and paste of cells with multiline text within the grid and between other applications is improved.

### Improved : changed TInifile use by TMemInifile for faster operation

Internal change to make persisting grid settings faster

### Improved : OLE file drop is now also possible on non cell area of grid

Previously, OLE drop was only possible on a grid cell. Now it is possible to drop files via OLE drop also on non cell areas in the grid.

### Improved : HTML form cell editing

Improved position control of editors on HTML forms

### Improved : painting of inplace edit controls on Windows Vista/Windows 7

General improvements in painting for Windows Vista & Windows 7

# What's new & improved in version v4.6

### New : TAdvGridImportDialog.Delimiter / CustomDelimiter properties added
With these new properties it is now possible to programmatically preset the preferred delimiter to start the CSV import dialog with and access the custom delimiter character.

### New : pressing Ctrl-Return on searchfooter will perform a backward search
Pressing return performs a forward search (equivalent to clicking the Find Next button) and the functionality has been added to let Ctrl-Return perform backward search (equivalent to clicking the Find Previous button)

### New : OnFileProgress triggered during SaveToXML()
The event is triggered while performing the export to XML to indicate the progress of the export.

### New : support for export of multi image cells to HTML
Saving a grid to file where for one or more cells multiple imagelist images were inserted now also outputs the different images to a file and links the images from the HTML table content.

### New : edValidChars inplace editor type
This new editor type allows to specify through the new property grid.ValidChars the series of characters accepted for input in the grid. If the grid.ValidChars property is set to '02468' for example, only characters '0', '2', '4', '6', '8' will be accepted for entry when the editor type is set to edValidChars.

### New : AutoFitColumns can be called with new parameter DoFixedCells
Calling AutoFitColumns applied an algorithm to change the column widths proportionally to fill the entire available width of the grid. The new default parameter DoFixedCells is set to true, ensuring that the proportional width change applies to both normal and fixed cells. When setting the parameter to false, only the width of normal columns is changed.

### New : Function SwapCells() added
This function swaps both cell content and cell properties of two cells.
Example

```
Grid.SwapCells(GridCoord(Col1,Row1), GridCoord(Col2,Row2));
```

### New : Lookup functionality added for edEditBtn editor type
Via the grid.LookupItems stringlist, values can be preset with which the inplace editor performs a lookup while typing. This was applicable for the edNormal and edCombo inplace editor formerly but has now been extended for the edEditBtn inplace editor as well.

### New : FilterDropDownRow property added to control on what fixed row filter dropdown appears
Via this property, it can now be controlled on what fixed row the filter dropdown button appears. Note that the filter dropdown button should appear on a fixed row and as such, grid.FilterDropDownRow should be smaller than grid.FixedRows

### New : SearchFooter.SearchDirection property added to control direction of search from searchfooter
With this new property it can be controlled whether the search performed from the search footer is from top to bottom row first and then from left to right, or from left to right first and then from top to bottom.

### New : Public property grid.PrintSettings.BorderColor added
This new public property allows to set the global grid border color for printing.

### Improved : export to HTML with special characters
Now special characters such as ö, é, â, .. are properly exported to HTML generating the HTML encoded sequence for such special characters.

### Improved : single row to multirow / single col to multicolumn smart clipboard handling
Now smart clipboard handling also works when row selection is active and selection is changed.

### Improved : small improvement wrt MouseActions.RangeSelectAndEdit = true option
Improved behaviour to start editing on non selected cells in the mode RangeSelectAndEdit = true

### Improved : drawing of cell pictures when BidiMode is RightToLeft
Cells with pictures now render correct with BidiMode is set to RightToLeft.

### Improved : display of sort indicator in rotated header cells
Now header cells can have rotated text as well as the sort indicator and this will paint correct when combined in a single header cell.


# What's new & improved in version v4.5

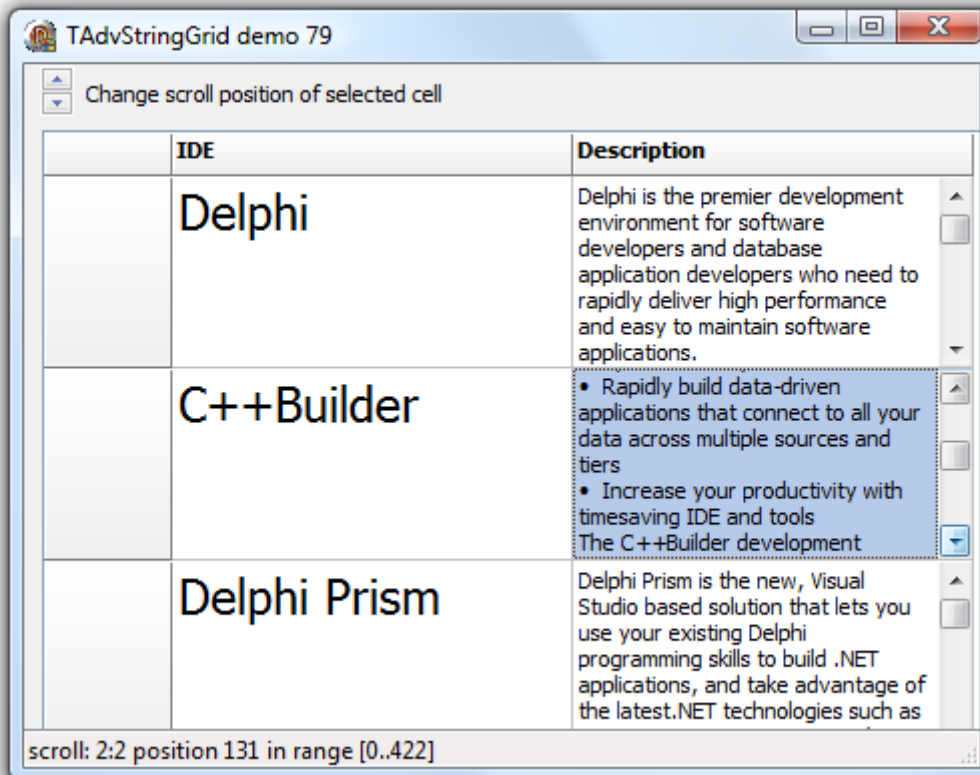**New : method RemoveSelectedCols, RemoveUnSelectedCols added**
With one call, the selected columns or unselected columns can now be removed from the grid. The columns can either be selected when range selection is enabled (goRangeSelect is true under grid.Options) or disjunct column selection is enabled (grid.MouseActions.DisjunctColSelect is true)

**New : easy way to persist & restore indexed sort settings**
Storing and restoring the latest state of how the user sorted one or more columns can now be easily persisted as a string value. This string value can be easily saved to the registry, INI file, XML file or a database. To save the sort state, retrieve this via grid.SortSettings.SaveToString. To restore the sort state, grid.SortSettings.LoadFromString(value: string) can used.

**New : scrollbars per cell**
Now it is possible to add a vertical scrollbar on cell level. The scrollbar can be defined as auto ranging scrollbar, ie. it will automatically adapt to the size of the text in the cell. Adding a scrollbar is done with grid.AddScrollBar(col,row: integer);



**New : added support for SUMMARY attribute for HTML export**
The HTMLSettings class has a new property Summary: string. With this property, the HTML TABLE summary attribute can be set. This attribute will be saved to the HTML file when used.

**New : ShowSeconds property added for TAdvStringGrid.SpinEdit**
Now it is possible to specify whether the time cell editor shows seconds or not. This can be set with the property grid.SpinEdit.ShowSeconds: Boolean.

**New : Navigation.AdvanceAutoEdit property added**

This new property determines whether the next cell selected when return is pressed when grid.Navigation.AdvanceOnEnter is set true will be automatically entered in edit mode or not.

**New : Navigation.ClipboardPasteAction property added**

This property allows to select what will happen during a paste in the grid. ClipboardPasteAction can be either set to paInsert or paReplace. When paReplace is selected, upon pasting, existing cell values will be replaced with the new values pasted. When paInsert is selected, upon pasting, the new values will be inserted in the grid at the coordinate of the selected cell.
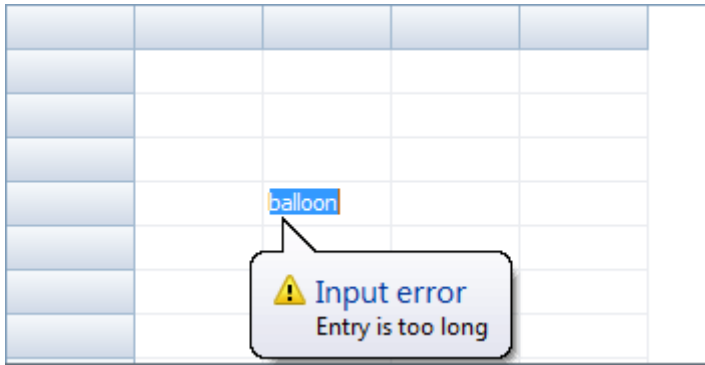
**New : Filtering on date+time**

In a filter condition it is now also possible to specify both a date and time. This filter condition will show all cells with a date/time value newer than Januari 1, 2009 12:00:

```
with grid.Filter.Add do
begin
  Column := 1;
  Condition := '> 1/1/2009 12:00';
end;
```

**New : Invalid entry icons**

New properties InvalidEntryTitle, InvalidEntryText, InvalidEntryIcon are added to enable showing a balloon informing when an invalid value was entered in the grid. This can be set from the grid.OnCellValidate event to signal the user invalid entries.
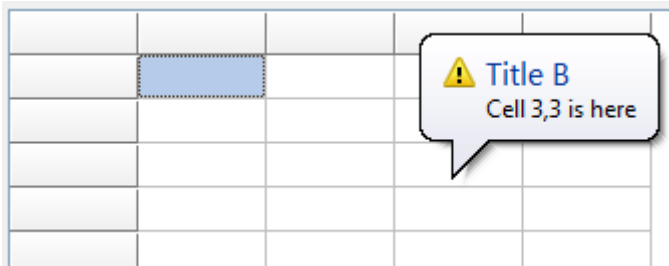
```
procedure TForm2.AdvStringGrid1CellValidate(Sender: TObject; ACol,
  ARow: Integer; var Value: string; var Valid: Boolean);
begin
  if length(value) < 3 then
  begin
    Advstringgrid1.InvalidEntryTitle := 'Input error';
    Advstringgrid1.InvalidEntryText := 'Entry not sufficiently long';
    Valid := false;
  end;
  if length(value) > 5 then
  begin
    Advstringgrid1.InvalidEntryTitle := 'Input error';
    Advstringgrid1.InvalidEntryText := 'Entry is too long';
    Valid := false;
  end;
end;
```

### New : AddBalloon, RemoveBalloon methods added

With the new methods grid.AddBalloon / grid.RemoveBalloon it is now possible to easily add a balloon to a grid cell. The balloon is displayed when the mouse hovers the cell. Via the grid.AddBalloon method, the balloon's title, text and icon can be set.

```
procedure TForm2.FormCreate(Sender: TObject);
begin
  AdvStringGrid1.Balloon.Enable := true;
  AdvStringGrid1.AddBalloon(2,2,'Title A','Cell 2,2 is here', biError);
  AdvStringGrid1.AddBalloon(3,3,'Title B','Cell 3,3 is here', biWarning);
end;
```



### New : filter dropdown in column header

It is now possible to enable the selection of a column filter at runtime by clicking a filter icon in the column header and upon selecting the filter, it is applied. The filter dropdown list can contain friendly filter names that are translated to filter conditions from the OnFilterSelect event. The filter dropdown items are set via the event OnGetColumnFilter:

```
procedure TForm2.AdvStringGrid1GetColumnFilter(Sender: TObject; Column: Integer; Filter: TStrings);

begin
  case Column of
  1:
    begin
      Filter.Add('Clear');
      Filter.Add('Within range');
      Filter.Add('Exceptions');
    end;
```

```
  2:
    begin
      Filter.Add('Clear');
      Filter.Add('>50');
      Filter.Add('<50');
    end;

  3:
    begin
      Filter.Add('Clear');
      Filter.Add('>20');
      Filter.Add('<20');
    end;

  4:
    begin
      Filter.Add('Clear');
      Filter.Add('>20');
      Filter.Add('<20');
    end;

  end;
end;
```

| | Within range | >50 | >20 | <20 |
|---|---|---|---|---|
| | 27 | Clear / Within range / Exceptions | 31 | 16 |
| | 49 | | 82 | 2 |
| | 34 | 81 | 81 | 14 |
| | 59 | 88 | 66 | 11 |
| | 76 | 98 | 49 | 3 |
| | 48 | 99 | 23 | 19 |
| | 43 | 68 | 75 | 3 |
| | 61 | 80 | 46 | 16 |

**New : incremental filtering & narrow down capability**
With the methods grid.ApplyFilter, grid.RemoveLastFilter, it is now possible to perform filtering in several steps and undo filter operations one by one. As such, two filter operations can be set after each other with:

```
with grid.Filter.Add do
begin
  Condition := 'condition1';
  Column := column1;
end;
grid.ApplyFilter; // performs the first filter operation

with grid.Filter.Add do
begin
  Condition := 'condition2';
  Column := column2;
```

```
end;
grid.ApplyFilter; // performs the second filter operation
```

and at a later time, the filters can be removed again one by one by calling:

```
grid.RemoveLastFilter;
grid.RemoveLastFilter;
```

Another new features is the narrow-down filter capability. The narrow-down filter can operate on every column in the grid or on a single column. A narrow-down can be performed by calling:

grid.NarrowDown(value:string [,optional column index]);

When calling grid.NarrowDown, the grid will show all rows that contain the string 'value' or all rows where the specified column contains the string 'value'. Successive calls to NarrowDown will perform less or more filtering with respect to the last narrow down call.

### New : filtering on full row

When it is not known in what column a value can be found and you want that the grid returns all rows that have any column with a specific value, the new filter Data type can be used: fcRow.

This filter will show all rows where any of its columns contains the word "text":

```
with grid.Filter.Add do
begin
  Condition := 'text';
  Data := fcRow;
end;
grid.FilterActive := true;
```

### New : Windows 7 & Office 2007 selection styles

Via grid.Look, two new styles can be selected: the Windows 7 style and the Office 2007 style. The Windows 7 style uses the Windows Explorer color scheme and applies the selection gradient identical to the selected items color in the Windows Explorer. The Office 2007 color scheme emulates the Office 2007 Luna color with the orange gradient selection colors.

Windows 7 color scheme



Office 2007 color scheme



**New : Windows Vista / Windows 7 Explorer row selection style**
With the new Windows Vista, Windows 7 Explorer row selection style, the entire selected row(s) have a single border.

**New : OnGetCellGradient event added**
Via the OnGetCellGradient event, a single or dual mirror gradient can be dynamically specified per cell.

**New : overriding fixed cell colors simplified**

Where in former versions it was required to select a single color style for fixed cells to override the color, now it is possible to set a fixed cell color for any type of fixed cell color (ie. solid color, single gradient, dual gradient)

### New : Replace function added

The replace function allows to replace text in cells. The specification for choosing in what cells to replace text is identical to the grid.Find() capabilities. With this sample code, for all cells that start with character '3' in column 2, the '3' will be replaced by 'A':
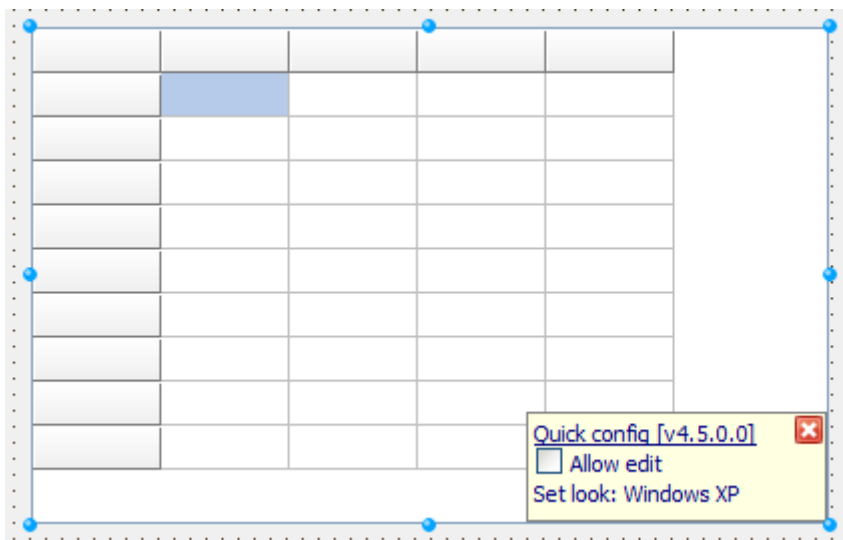
```
AdvStringGrid1.Col := 2;
AdvStringgrid1.Replace('3','A',[fnFindInCurrentCol, fnMatchStart]);
```

### New : ColumnSize.SynchNormalCellsOnly property added

When grid.ColumnSize.SynchWithGrid is set true, the size of all columns will proportionally change with grid width changes. It can be desirable that fixed columns are excluded for this proportional sizing of columns. This can be enabled now by setting grid.ColumnSize.SynchNormalCellOnly to True.

### New : Quick config design time helper

At design time, a quick config box is shown by default. This provides easy & fast access to enable or disable editing in the grid as well as make a selection from the different built-in looks for the grid:



### New : HTMLSettings.ExportImages property added

Via the grid.HTMLSettings.ExportImages property it can be controlled whether images are exported or not when saving the grid to HTML.

### Improved : custom inplace editor usage

Positioning and handling of auto advance, edit cancelling has been improved when custom inplace edit controls are used via TEditLink classes.

**Improved : inplace date editor starts with first char entered**
Now the date editor is also started and entered in edit mode with the first character pressed on the grid to start the editing.

**Improved : mouse handling when PreciseCheckBoxCheck = true**
For a checkbox in a right-aligned cell, the PreciseCheckBoxCheck will now also work to accept a checkbox click only when the checkbox is clicked and not the entire cell.

**Improved : position of text during edit versus display**
Position of text in inplace editors now 100% matches the position of the text in the cell when in normal display mode.

**Improved : export to HTML**
Export to HTML now handles merged cells and hidden columns better.

**Improved : handling of cell validation**
Cell validation and keeping the inplace editor active when incorrect values are entered is now more consistent for the various supported inplace editor types.

**Improved : faster handling of row hiding / node expand & collaps**
The performance of the algorithm to hide rows, expand and collaps nodes has been improved.

**Improved : button painting in SearchFooter**
The buttons in the search footer now properly paint with Windows theme when enabled.

**Improved : shortcut handling in search footer**
Accelerator keys on search footer buttons are now handled.

**Improved : automatic sort type detection**
Algorithm to automatically detect whether a date, a time or a combined date & time is in a cell has been improved.

**Fixed : issue with mouse wheel & disjunct row selection**
Issue with selected rows being affected by using the mousewheel with the grid has been fixed.