

EAdvGridError

Hierarchy _____

Unit

advgrid

EAdvGridError=**Class**(Exception);

EAdvGridError Events

[EAdvGridError](#)

[Alphabetically](#)

[Legend](#)

EAdvGridError Events

[EAdvGridError](#)

[.By object](#)

[Legend](#)

EAdvGridError Methods

[EAdvGridError](#)

[Alphabetically](#)

[Legend](#)

EAdvGridError Methods

[EAdvGridError](#)

[.By object](#)

[Legend](#)

EAdvGridError Properties

[EAdvGridError](#)

[Alphabetically](#)

[Legend](#)

EAdvGridError Properties

[EAdvGridError](#)

[.By object](#)

[Legend](#)

Globals

Properties

BLD_VER

Globals

Unit

advgrid

```
BLD_VER=0;
```

CF_GRIDCELLS

Globals

Unit

advgrid

CF_GRIDCELLS: **Word**;

MAJ_VER

Globals

Unit

advgrid

MAJ_VER=2;

MAXCOLUMNS

[Globals](#)

Unit

advgrid

MAXCOLUMNS=512;

MIN_VER

Globals

Unit

advgrid

MIN_VER=3;

REL_VER

Globals

Unit

advgrid

REL_VER=0;

RTF_TWIPS

Globals

Unit

advgrid

RTF_TWIPS=1440;

Globals Properties

[Globals](#) [Alphabetically](#)

[Legend](#)

In Globals

[BLD_VER](#)
[CF_GRIDCELLS](#)
[MAJ_VER](#)
[MAXCOLUMNS](#)
[MIN_VER](#)
[REL_VER](#)
[RTF_TWIPS](#)

Globals Properties

[Globals](#) [By object](#)

[Legend](#)

[BLD_VER](#)
[CF_GRIDCELLS](#)
[MAJ_VER](#)
[MAXCOLUMNS](#)
[MIN_VER](#)
[REL_VER](#)
[RTF_TWIPS](#)

PBoolArray

Unit

advgrid

PBoolArray=^TBoolarray;

TAdvanceDirection

Unit

advgrid

TAdvanceDirection=(adLeftRight, adTopBottom); **Description**

Value

Meaning

adLeftRight direction goes from left to right first, then from top to bottom

adTopBottom direction goes from top to bottom first, then from left to right

TAdvInplaceEdit

[Hierarchy](#)

[Properties](#)

[Methods](#)

[Events](#)

Unit

advgrid

Description

Base inplace editor for TAdvStringGrid

TAdvInplaceEdit.DoChange

[TAdvInplaceEdit](#)

procedure DoChange;

TAdvInplaceEdit.LengthLimit

[TAdvInplaceEdit](#)

property LengthLimit: **smallint**; **Description**

Sets the max. editable length of the inplace editor.

TAdvInplaceEdit.VAlign

[TAdvInplaceEdit](#)

property VAlign: Boolean; **Description**

Sets the vertical alignment of the inplace editor.

TAdvInplaceEdit.WordWrap

[TAdvInplaceEdit](#)

property WordWrap: Boolean; **Description**

Sets if the inplace editor should perform wordwrap or not.

TAdvInplaceEdit Events

[TAdvInplaceEdit](#) [Alphabetically](#)

[Legend](#)

Derived from TCustomEdit

Derived from TWinControl

Derived from TControl

TAdvInplaceEdit Events

[TAdvInplaceEdit](#)

[.By object](#)

[Legend](#)

TAdvInplaceEdit Methods

[TAdvInplaceEdit](#) [Alphabetically](#) [Legend](#)

In TAdvInplaceEdit

[DoChange](#)

Derived from TInplaceEdit

[Create](#)
[Deselect](#)
[Hide](#)
[Invalidate](#)
[Move](#)
[PosEqual](#)
[SetFocus](#)
[UpdateLoc](#)
[Visible](#)

Derived from TCustomMaskEdit

[Clear](#)
[GetTextLen](#)
[ValidateEdit](#)

Derived from TCustomEdit

[ClearSelection](#)
[ClearUndo](#)
[CopyToClipboard](#)
[CutToClipboard](#)
[DefaultHandler](#)
[GetSelTextBuf](#)
[PasteFromClipboard](#)
[SelectAll](#)
[SetSelTextBuf](#)
[Undo](#)

Derived from TWinControl

[Broadcast](#)
[CanFocus](#)
[ContainsControl](#)
[ControlAtPos](#)
[CreateParented](#)
[Destroy](#)
[DisableAlign](#)
[DockDrop](#)
[EnableAlign](#)
[FindChildControl](#)
[FlipChildren](#)
[Focused](#)
[GetTabOrderList](#)
[HandleAllocated](#)
[HandleNeeded](#)
[InsertControl](#)
[PaintTo](#)
[Realign](#)
[RemoveControl](#)
[Repaint](#)
[ScaleBy](#)
[ScrollBy](#)
[SetBounds](#)
[Update](#)
[UpdateControlState](#)

Derived from TControl

[BeginDrag](#)
[BringToFront](#)
[ClientToParent](#)
[ClientToScreen](#)
[Dock](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EndDrag](#)
[GetControlsAlignment](#)
[GetParentComponent](#)
[GetTextBuf](#)
[HasParent](#)
[InitiateAction](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[ParentToClient](#)
[Perform](#)
[Refresh](#)
[ReplaceDockedControl](#)
[ScreenToClient](#)
[SendToBack](#)
[SetTextBuf](#)
[Show](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

Derived from TComponent

[BeforeDestruction](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TAdvInplaceEdit Methods

TAdvInplaceEdit By object Legend

Assign
BeforeDestruction
BeginDrag
BringToFront
Broadcast
CanFocus
Clear
ClearSelection
ClearUndo
ClientToParent
ClientToScreen
ContainsControl
ControlAtPos
CopyToClipboard
Create
CreateParented
CutToClipboard
DefaultHandler
Deselect
Destroy
DestroyComponents
Destroying
DisableAlign
DoChange
Dock
DockDrop
DragDrop
Dragging
DrawTextBiDiModeFlags
DrawTextBiDiModeFlagsReadingOn
EnableAlign
EndDrag
ExecuteAction
FindChildControl
FindComponent
FlipChildren
Focused
FreeNotification
FreeOnRelease
GetControlsAlignment
GetNamePath
GetParentComponent
GetSelTextBuf
GetTabOrderList
GetTextBuf
GetTextLen
HandleAllocated
HandleNeeded
HasParent
Hide
InitiateAction
InsertComponent
InsertControl
Invalidate
IsImplementorOf
IsRightToLeft
ManualDock
ManualFloat

[MouseWheelHandler](#)
[Move](#)
[PaintTo](#)
[ParentToClient](#)
[PasteFromClipboard](#)
[Perform](#)
[PosEqual](#)
[Realign](#)
[ReferenceInterface](#)
[Refresh](#)
[RemoveComponent](#)
[RemoveControl](#)
[RemoveFreeNotification](#)
[Repaint](#)
[ReplaceDockedControl](#)
[SafeCallException](#)
[ScaleBy](#)
[ScreenToClient](#)
[ScrollBy](#)
[SelectAll](#)
[SendToBack](#)
[SetBounds](#)
[SetFocus](#)
[SetSelTextBuf](#)
[SetSubComponent](#)
[SetTextBuf](#)
[Show](#)
[Undo](#)
[Update](#)
[UpdateAction](#)
[UpdateControlState](#)
[UpdateLoc](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)
[ValidateEdit](#)
[Visible](#)

TAdvInplaceEdit Properties

[TAdvInplaceEdit](#) [Alphabetically](#) [Legend](#)

In TAdvInplaceEdit

- [LengthLimit](#)
- [VAlign](#)
- [WordWrap](#)

Derived from TInplaceEdit

Derived from TCustomMaskEdit

- ▶ [EditText](#)
- ▶ [IsMasked](#)
- ▶ [Text](#)

Derived from TCustomEdit

- ▶ [CanUndo](#)
- ▶ [Modified](#)
- ▶ [SelLength](#)
- ▶ [SelStart](#)
- ▶ [SelText](#)
- [TabStop](#)

Derived from TWinControl

- ▶ [AlignDisabled](#)
- ▶ [Brush](#)
- ▶ [class](#)
- ▶ [ControlCount](#)
- ▶ [Controls](#)
- ▶ [DockClientCount](#)
- ▶ [DockClients](#)
- ▶ [DoubleBuffered](#)
- ▶ [Handle](#)
- ▶ [ParentWindow](#)
- ▶ [Showing](#)
- ▶ [TabOrder](#)
- ▶ [VisibleDockClientCount](#)

Derived from TControl

- ▶ [Action](#)
- ▶ [Align](#)
- ▶ [Anchors](#)
- ▶ [BiDiMode](#)
- ▶ [BoundsRect](#)
- ▶ [ClientHeight](#)
- ▶ [ClientOrigin](#)
- ▶ [ClientRect](#)
- ▶ [ClientWidth](#)
- ▶ [Constraints](#)
- ▶ [ControlState](#)
- ▶ [ControlStyle](#)
- [Cursor](#)
- [DockOrientation](#)
- [Enabled](#)
- ▶ [Floating](#)
- ▶ [FloatingDockSiteClass](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)

- [Left](#)
 - [LRDockWidth](#)
 - [Parent](#)
 - [ShowHint](#)
 - [TBDockHeight](#)
- [Top](#)
 - [UndockHeight](#)
 - [UndockWidth](#)
 - [Visible](#)
- [Width](#)
 - [WindowProc](#)

Derived from TComponent

- ▶ [ComObject](#)
- ▶ [ComponentCount](#)
- ▶ [ComponentIndex](#)
- ▶ [Components](#)
- ▶ [ComponentState](#)
- ▶ [ComponentStyle](#)
- ▶ [DesignInfo](#)
- [Name](#)
- ▶ [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TAdvInplaceEdit Properties

TAdvInplaceEdit By object Legend

- [Action](#)
- [Align](#)
- ▶ [AlignDisabled](#)
- [Anchors](#)
- [BiDiMode](#)
- [BoundsRect](#)
- ▶ [Brush](#)
- ▶ [CanUndo](#)
- [class](#)
- [ClientHeight](#)
- ▶ [ClientOrigin](#)
- ▶ [ClientRect](#)
- [ClientWidth](#)
- ▶ [ComObject](#)
- ▶ [ComponentCount](#)
- [ComponentIndex](#)
- ▶ [Components](#)
- ▶ [ComponentState](#)
- ▶ [ComponentStyle](#)
- [Constraints](#)
- ▶ [ControlCount](#)
- ▶ [Controls](#)
- [ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
- [DesignInfo](#)
- ▶ [DockClientCount](#)
- ▶ [DockClients](#)
- [DockOrientation](#)
- [DoubleBuffered](#)
- [EditText](#)
- [Enabled](#)
- ▶ [Floating](#)
- [FloatingDockSiteClass](#)
- ▶ [Handle](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- ▶ [IsMasked](#)
- [Left](#)
- [LengthLimit](#)
- [LRDockWidth](#)
- [Modified](#)
- [Name](#)
- ▶ [Owner](#)
- [Parent](#)
- [ParentWindow](#)
- [SelLength](#)
- [SelStart](#)
- [SelText](#)
- [ShowHint](#)
- ▶ [Showing](#)
- [TabOrder](#)
- [TabStop](#)
- [Tag](#)

TBDockHeight

Text

- Top
- UndockHeight
- UndockWidth
- VAlign
- VCLComObject
- Visible
- VisibleDockClientCount
- Width
- WindowProc
- WordWrap

TAdvRichEdit

[Hierarchy](#)

[Properties](#)

[Methods](#)

[Events](#)

Unit

advgrid

Description

TAdvStringGrid inplace rich text editor

TAdvRichEdit.SelNormal

[TAdvRichEdit](#)

procedure SelNormal; **Description**

Sets the selected text to normal vertical alignment.

TAdvRichEdit.SelSubscript

[TAdvRichEdit](#)

procedure SelSubscript; **Description**

Sets the selected text to subscript vertical alignment.

TAdvRichEdit.SelSuperscript

[TAdvRichEdit](#)

procedure SelSuperscript; **Description**

Sets the selected text to superscript vertical alignment.

TAdvRichEdit.ReqHeight

[TAdvRichEdit](#)

property ReqHeight: **Integer**; **Description**

Required height in pixels of current rich text contents

TAdvRichEdit.ReqWidth

[TAdvRichEdit](#)

property ReqWidth: **Integer**; **Description**

Required width in pixels of current rich text contents

TAdvRichEdit Events

[TAdvRichEdit](#) [Alphabetically](#)

[Legend](#)

Derived from TRichEdit

- [OnChange](#)
- [OnContextPopup](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnMouseWheel](#)
- [OnMouseWheelDown](#)
- [OnMouseWheelUp](#)
- [OnProtectChange](#)
- [OnResizeRequest](#)
- [OnSaveClipboard](#)
- [OnSelectionChange](#)
- [OnStartDock](#)
- [OnStartDrag](#)

Derived from TWinControl

Derived from TControl

TAdvRichEdit Events

[TAdvRichEdit](#)

[By object](#)

[Legend](#)

- [OnChange](#)
- [OnContextPopup](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnMouseWheel](#)
- [OnMouseWheelDown](#)
- [OnMouseWheelUp](#)
- [OnProtectChange](#)
- [OnResizeRequest](#)
- [OnSaveClipboard](#)
- [OnSelectionChange](#)
- [OnStartDock](#)
- [OnStartDrag](#)

TAdvRichEdit Methods

[TAdvRichEdit](#) [Alphabetically](#) [Legend](#)

In TAdvRichEdit

[SelNormal](#)
[SelSubscript](#)
[SelSuperscript](#)

Derived from TCustomRichEdit

[Clear](#)
[Create](#)
[Destroy](#)
[FindText](#)
[GetSelTextBuf](#)
[Print](#)

Derived from TCustomMemo

[GetControlsAlignment](#)

Derived from TCustomEdit

[ClearSelection](#)
[ClearUndo](#)
[CopyToClipboard](#)
[CutToClipboard](#)
[DefaultHandler](#)
[PasteFromClipboard](#)
[SelectAll](#)
[SetSelTextBuf](#)
[Undo](#)

Derived from TWinControl

[Broadcast](#)
[CanFocus](#)
[ContainsControl](#)
[ControlAtPos](#)
[CreateParented](#)
[DisableAlign](#)
[DockDrop](#)
[EnableAlign](#)
[FindChildControl](#)
[FlipChildren](#)
[Focused](#)
[GetTabOrderList](#)
[HandleAllocated](#)
[HandleNeeded](#)
[InsertControl](#)
[Invalidate](#)
[PaintTo](#)
[Realign](#)
[RemoveControl](#)
[Repaint](#)
[ScaleBy](#)
[ScrollBy](#)
[SetBounds](#)
[SetFocus](#)
[Update](#)
[UpdateControlState](#)

Derived from TControl

[BeginDrag](#)
[BringToFront](#)
[ClientToParent](#)

[ClientToScreen](#)
[Dock](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EndDrag](#)
[GetParentComponent](#)
[GetTextBuf](#)
[GetTextLen](#)
[HasParent](#)
[Hide](#)
[InitiateAction](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[ParentToClient](#)
[Perform](#)
[Refresh](#)
[ReplaceDockedControl](#)
[ScreenToClient](#)
[SendToBack](#)
[SetTextBuf](#)
[Show](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

Derived from TComponent

[BeforeDestruction](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TAdvRichEdit Methods

TAdvRichEdit By object Legend

Assign
BeforeDestruction
BeginDrag
BringToFront
Broadcast
CanFocus
Clear
ClearSelection
ClearUndo
ClientToParent
ClientToScreen
ContainsControl
ControlAtPos
CopyToClipboard
Create
CreateParented
CutToClipboard
DefaultHandler
Destroy
DestroyComponents
Destroying
DisableAlign
Dock
DockDrop
DragDrop
Dragging
DrawTextBiDiModeFlags
DrawTextBiDiModeFlagsReadingOn
EnableAlign
EndDrag
ExecuteAction
FindChildControl
FindComponent
FindText
FlipChildren
Focused
FreeNotification
FreeOnRelease
GetControlsAlignment
GetNamePath
GetParentComponent
GetSelTextBuf
GetTabOrderList
GetTextBuf
GetTextLen
HandleAllocated
HandleNeeded
HasParent
Hide
InitiateAction
InsertComponent
InsertControl
Invalidate
IsImplementorOf
IsRightToLeft
ManualDock
ManualFloat
MouseWheelHandler

[PaintTo](#)
[ParentToClient](#)
[PasteFromClipboard](#)
[Perform](#)
[Print](#)
[Realign](#)
[ReferenceInterface](#)
[Refresh](#)
[RemoveComponent](#)
[RemoveControl](#)
[RemoveFreeNotification](#)
[Repaint](#)
[ReplaceDockedControl](#)
[SafeCallException](#)
[ScaleBy](#)
[ScreenToClient](#)
[ScrollBy](#)
[SelectAll](#)
[SelNormal](#)
[SelSubscript](#)
[SelSuperscript](#)
[SendToBack](#)
[SetBounds](#)
[SetFocus](#)
[SetSelTextBuf](#)
[SetSubComponent](#)
[SetTextBuf](#)
[Show](#)
[Undo](#)
[Update](#)
[UpdateAction](#)
[UpdateControlState](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

TAdvRichEdit Properties

[TAdvRichEdit](#) [Alphabetically](#) [Legend](#)

In TAdvRichEdit

-
- [ReqHeight](#)
-
- [ReqWidth](#)

Derived from TRichEdit

- [Align](#)
- [Alignment](#)
- [Anchors](#)
- [BevelEdges](#)
- [BevelInner](#)
- [BevelKind](#)
- [BevelOuter](#)
- [BevelWidth](#)
- [BiDiMode](#)
- [BorderStyle](#)
- [BorderWidth](#)
- [Color](#)
- [Constraints](#)
- [Ctl3D](#)
- [DragCursor](#)
- [DragKind](#)
- [DragMode](#)
- [Enabled](#)
- [Font](#)
- [HideScrollBars](#)
- [HideSelection](#)
- [ImeMode](#)
- [ImeName](#)
- [Lines](#)
- [MaxLength](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentCtl3D](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [PlainText](#)
- [PopupMenu](#)
- [ReadOnly](#)
- [ScrollBars](#)
- [ShowHint](#)
- [TabOrder](#)
- [TabStop](#)
- [Visible](#)
- [WantReturns](#)
- [WantTabs](#)
- [WordWrap](#)

Derived from TCustomRichEdit

- [AConversionClass](#)
- [class](#)
- [DefAttributes](#)
- [DefaultConverter](#)
- [PageRect](#)
- [Paragraph](#)
- [SelAttributes](#)

Derived from TCustomMemo

[CaretPos](#)

Derived from TCustomEdit

- [CanUndo](#)
- [Modified](#)
- [SelLength](#)
- [SelStart](#)
- [SelText](#)
- [Text](#)

Derived from TWinControl

- [AlignDisabled](#)
- [Brush](#)
- [ControlCount](#)
- [Controls](#)
- [DockClientCount](#)
- [DockClients](#)
- [DoubleBuffered](#)
- [Handle](#)
- [ParentWindow](#)
- [Showing](#)
- [VisibleDockClientCount](#)

Derived from TControl

- [Action](#)
- [BoundsRect](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
- [DockOrientation](#)
- [Floating](#)
- [FloatingDockSiteClass](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LRDockWidth](#)
- [Parent](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [Width](#)
- [WindowProc](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)

- [Tag](#)
[VCLComObject](#)

TAdvRichEdit Properties

TAdvRichEdit By object Legend

AConversionClass

Action

- Align
- AlignDisabled
- Alignment
- Anchors
- BevelEdges
- BevelInner
- BevelKind
- BevelOuter
- BevelWidth
- BiDiMode
- BorderStyle
- BorderWidth

BoundsRect

- Brush
- CanUndo
- CaretPos
- class
- ClientHeight
- ClientOrigin
- ClientRect
- ClientWidth
- Color
- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- Constraints
- ControlCount
- Controls
- ControlState

ControlStyle

- Ctl3D
- Cursor
- DefAttributes
- DefaultConverter
- DesignInfo
- DockClientCount
- DockClients
- DockOrientation
- DoubleBuffered
- DragCursor
- DragKind
- DragMode
- Enabled
- Floating
- FloatingDockSiteClass
- Font
- Handle
- Height
- HelpContext
- HelpKeyword
- HelpType
- HideScrollBars
- HideSelection

- [Hint](#)
- [HostDockSite](#)
- [ImeMode](#)
- [ImeName](#)
- [Left](#)
- [Lines](#)
- [LRDockWidth](#)
- [MaxLength](#)
- [Modified](#)
- [Name](#)
- [Owner](#)
- [PageRect](#)
- [Paragraph](#)
- [Parent](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentCtl3D](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [ParentWindow](#)
- [PlainText](#)
- [PopupMenu](#)
- [ReadOnly](#)
- [ReqHeight](#)
- [ReqWidth](#)
- [ScrollBars](#)
- [SelAttributes](#)
- [SelLength](#)
- [SelStart](#)
- [SelText](#)
- [ShowHint](#)
- [Showing](#)
- [TabOrder](#)
- [TabStop](#)
- [Tag](#)
- [TBDockHeight](#)
- [Text](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [VCLComObject](#)
- [Visible](#)
- [VisibleDockClientCount](#)
- [WantReturns](#)
- [WantTabs](#)
- [Width](#)
- [WindowProc](#)
- [WordWrap](#)



TAdvStringGrid

[Hierarchy](#)

[Properties](#)

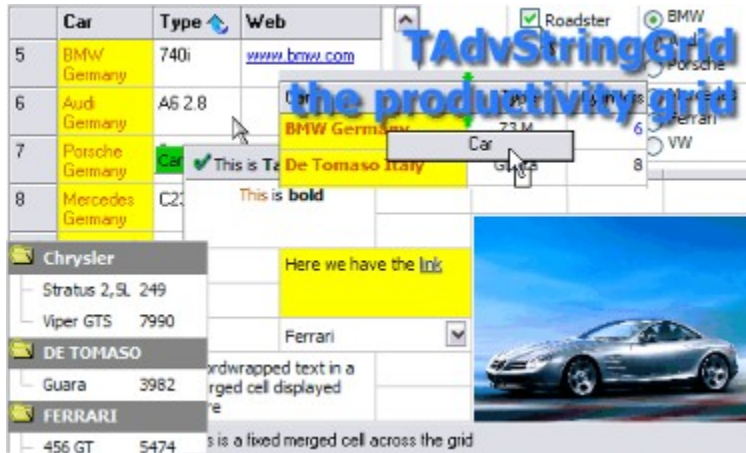
[Methods](#) [Events](#)

Unit

advgrid

Description

TAdvStringGrid v 2.3



Copyright © 1996-2002 by TMS software

Web: <http://www.tmssoftware.com>

Email : <mailto:info@tmssoftware.com>

Single developer license for commercial use, source code and full version cycle free updates : 60 USD

Also available :

- TMS Component pack : includes TAdvStringGrid
- TAdvSpreadGrid : spreadsheet add-on
- TAdvColumnGrid : TAdvStringGrid extended with column property management

TAdvStringGrid.OnAnchorClick

[TAdvStringGrid](#)

Event handler is called when an anchor within a HTML formatted cells is clicked.

TAnchorClickEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; Anchor: **string**; var AutoHandle: Boolean) **of object**;

property OnAnchorClick: [TAnchorClickEvent](#); **Description**

The anchor parameter is the value of the HREF tag. Set AutoHandle to true if the grid can perform a ShellExecute on the anchor, or set to false and handle the anchorclick from the application.

TAdvStringGrid.OnAnchorEnter

[TAdvStringGrid](#)

TAnchorEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; Anchor: **string**) **of object**;

property OnAnchorEnter: [TAnchorEvent](#); **Description**

Event called when the mouse enters an anchor inside a cell with HTML formatted text.

TAdvStringGrid.OnAnchorExit

[TAdvStringGrid](#)

TAnchorEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; Anchor: **string**) **of object**;

property OnAnchorExit: [TAnchorEvent](#); **Description**

Event called when the mouse leaves an anchor inside a cell with HTML formatted text.

TAdvStringGrid.OnAnchorHint

[TAdvStringGrid](#)

TAnchorHintEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; var Anchor: **string**) of object;

property OnAnchorHint: [TAnchorHintEvent](#); **Description**

Event triggered when a hint is displayed for an anchor in a HTML formatted cell

TAdvStringGrid.OnAutoAddRow

[TAdvStringGrid](#)

Event triggered when a row is added when navigating to the end of the grid and Navigation.AdvanceInsert is true.

TAutoAddRowEvent=**procedure**(Sender: TObject; ARow: **Integer**) **of object**;

property OnAutoAddRow: [TAutoAddRowEvent](#); **Description**

TNavigation.AdvanceInsert must be true to allow automatic row insertion after navigation through the last row.

TAdvStringGrid.OnAutoDeleteRow

[TAdvStringGrid](#)

Event handler called after user has deleted a row by pressing DEL button.

TAutoDeleteRowEvent=**procedure**(Sender: TObject; ARow: **Integer**) **of object**;

property OnAutoDeleteRow: [TAutoDeleteRowEvent](#); **Description**

The Arow parameter indicates the row that is deleted

TAdvStringGrid.OnAutoInsertCol

[TAdvStringGrid](#)

Event triggered when a column is added when navigating to the end of the grid and when Navigation.AdvanceInsert is true.

TAutoInsertColEvent=**procedure**(Sender: TObject; ACol: **Integer**) **of object**;

property OnAutoInsertCol: [TAutoInsertColEvent](#); **Description**

TNavigation.AdvanceInsert must be true to allow automatic column insertion after navigation through the last column.

TAdvStringGrid.OnAutoInsertRow

[TAdvStringGrid](#)

Event handler called after user has inserted a row by pressing INS button. Can be used to initialize a row to default data.

TAutoInsertRowEvent=**procedure**(Sender: TObject; ARow: **Integer**) **of object**;

property OnAutoInsertRow: [TAutoInsertRowEvent](#); **Description**

The Arow parameter indicates the row insert position

TAdvStringGrid.OnButtonClick

[TAdvStringGrid](#)

TButtonClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**) of object;

property OnButtonClick: [TButtonClickEvent](#); **Description**

Event fired when a button in the cell is pressed. The button is added to the grid with the AddButton method.

TAdvStringGrid.OnCanAddRow

[TAdvStringGrid](#)

TCanAddRowEvent=**procedure**(Sender: TObject; **var** CanAdd: Boolean) **of object**;

property OnCanAddRow: [TCanAddRowEvent](#); **Description**

Event triggered to query if a new row can be added when navigating from the current last row of the grid.

TAdvStringGrid.OnCanDeleteRow

[TAdvStringGrid](#)

TCanDeleteRowEvent=**procedure**(Sender: TObject; ARow: **Integer**; var CanDelete: Boolean) **of object**;

property OnCanDeleteRow: [TCanDeleteRowEvent](#); **Description**

Event triggered to query if a row can be deleted with the Del key.

TAdvStringGrid.OnCanEditCell

[TAdvStringGrid](#)

Event handler to set a cell to read-only.

TCanEditCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; **var** CanEdit: Boolean) **of object**;

property OnCanEditCell: [TCanEditCellEvent](#); **Description**

Set canedit to false if the cell Arow,Acoll must be readonly. The OnCanEditCell can be used for different types of inplace editors

TAdvStringGrid.OnCanInsertRow

[TAdvStringGrid](#)

TCanInsertRowEvent=**procedure**(Sender: TObject; ARow: **Integer**; **var** CanInsert: Boolean) **of object**;

property OnCanInsertRow: [TCanInsertRowEvent](#); **Description**

Event triggered to query if a row can be inserted with the Ins key.

TAdvStringGrid.OnCanSort

[TAdvStringGrid](#)

Event handler called when user clicked on a column header to sort a column.

TCanSortEvent=**procedure**(Sender: TObject; ACol: **Integer**; var DoSort: Boolean) **of object**;

property OnCanSort: [TCanSortEvent](#); **Description**

The event handler is called before starting the sort to query if the column can be sorted. If no handler is specified or handler simply returns, sorting is performed, otherwise, set dosort to false to disable sorting a specific column. The Acol parameter indicates which column will be sorted. This event is called before the sort starts. Only when dosort is true, the sort starts.

TAdvStringGrid.OnCellChanging

[TAdvStringGrid](#)

Event handler is called when the selected row is about to change.

TCellChangingEvent=**procedure**(Sender: TObject; OldRow, OldCol, NewRow, NewCol: **Integer**; var Allow: Boolean) **of object**;

property OnCellChanging: [TCellChangingEvent](#); **Description**

Called when the grid's cell changes from the OldRow,OldCol to NewRow,NewCol. With the Allow parameter, this change can be programmatically disabled or not.

TAdvStringGrid.OnCellsChanged

[TAdvStringGrid](#)

TCellsChangedEvent=**procedure**(Sender: TObject; R: TRect) **of object**;

property OnCellsChanged: [TCellsChangedEvent](#); **Description**

Event triggered whenever the contents of a cell change. Cell contents can change due to editing, clipboard operations or loading new contents from file.

TAdvStringGrid.OnCellValidate

[TAdvStringGrid](#)

Event handler to add possibility to do input validation or autocorrection.

TCellValidateEvent=**procedure**(Sender: TObject; Col, Row: **Integer**; var Value: **String**; var Valid: Boolean) **of object**;

property OnCellValidate: [TCellValidateEvent](#); **Description**

The event handler is called after the user leaves a cell after editing. Set the Valid parameter to false if input is not accepted and cell should contain old value.

For autocorrection, the edited string, passed in the Value parameter can be modified while the Valid parameter is true. The corrected value will be entered in the grid instead of the typed value.

TAdvStringGrid.OnCheckBoxClick

[TAdvStringGrid](#)

Event handler called after the current checkbox inplace editor is clicked.

TCheckBoxClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; State: Boolean) **of object**;

property OnCheckBoxClick: [TCheckBoxClickEvent](#); **Description**

aCol and aRow indicate the cell in which the checkbox is displayed and the state parameter indicates the new checkbox state.

TAdvStringGrid.OnCheckBoxMouseUp

[TAdvStringGrid](#)

TCheckBoxClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; State: Boolean) **of object**;

property OnCheckBoxMouseUp: [TCheckBoxClickEvent](#); **Description**

Event triggered on MouseUp event for a checkbox.

TAdvStringGrid.OnClickCell

[TAdvStringGrid](#)

Event handler called when left mouse is pressed on a cell which returns column and row of the selected cell.

TClickCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**) **of object**;

property OnClickCell: [TClickCellEvent](#); **Description**

ARow and ACol indicate the clicked cell.

TAdvStringGrid.OnClickSort

[TAdvStringGrid](#)

Event handler called when user clicked on the fixed column header to sort on a specific column.

TClickSortEvent=**procedure**(Sender: TObject; ACol: **Integer**) of object;

property OnClickSort: [TClickSortEvent](#); **Description**

The Acol parameter indicates which column is sorted. This event is called when the sort is completed.

Can be used in conjunction with the OnCanSort event which is called before the sort starts.

TAdvStringGrid.OnClipboardBeforePasteCell

[TAdvStringGrid](#)

TBeforeCellPasteEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; var Value: **string**; var Allow: Boolean) **of object**;

property OnClipboardBeforePasteCell: [TBeforeCellPasteEvent](#);

Description

Event triggered before cell contents are updated through a paste. Changing the Value parameter allows dynamically changing the pasted cell contents and setting the Allow parameter to false allows preventing the paste to take place.

TAdvStringGrid.OnClipboardCopy

[TAdvStringGrid](#)

TClipboardEvent=**procedure**(Sender: TObject; **var** Allow: Boolean) **of object**;

property OnClipboardCopy: [TClipboardEvent](#); **Description**

Event triggered when the user performs a clipboard copy with keyboard shortcuts.

Use allow to enable / disable the clipboard action.

TAdvStringGrid.OnClipboardCut

[TAdvStringGrid](#)

TClipboardEvent=**procedure**(Sender: TObject; **var** Allow: Boolean) **of object**;

property OnClipboardCut: [TClipboardEvent](#); **Description**

Event triggered when the user performs a clipboard cut with keyboard shortcuts

Use allow to enable / disable the clipboard action.

TAdvStringGrid.OnClipboardPaste

[TAdvStringGrid](#)

TClipboardEvent=**procedure**(Sender: TObject; **var** Allow: Boolean) **of object**;

property OnClipboardPaste: [TClipboardEvent](#); **Description**

Event triggered when the user performs a clipboard paste with keyboard shortcuts.

Use allow to enable / disable the clipboard action.

TAdvStringGrid.OnColChanging

[TAdvStringGrid](#)

TColChangingEvent=**procedure**(Sender: TObject; OldCol, NewCol: **Integer**; var Allow: Boolean) of **object**;

property OnColChanging: [TColChangingEvent](#); **Description**

Called when the grid's column changes from the OldCol to NewCol. With the Allow parameter, this change can be programmatically disabled or not.

TAdvStringGrid.OnColumnMove

[TAdvStringGrid](#)

TColumnSizeEvent=**procedure**(Sender: TObject; ACol: **Integer**; var Allow: Boolean) **of object**;

property OnColumnMove: [TColumnSizeEvent](#); **Description**

Event triggered during move of a column. Column moving can be allowed or not through the Allow parameter.

TAdvStringGrid.OnColumnSize

[TAdvStringGrid](#)

TColumnSizeEvent=**procedure**(Sender: TObject; ACol: **Integer**; var Allow: Boolean) **of object**;

property OnColumnSize: [TColumnSizeEvent](#); **Description**

This allows to control programmatically which columns can be resized and which not rather than a global control with goColumnSizing.

goColumnSizing must be set true after which OnColumnSize is called on start of every column resizing with mouse.Allow is default true. To disable sizing of column aCol, set Allow to false.

TAdvStringGrid.OnComboChange

[TAdvStringGrid](#)

Event handler called whenever the selection of the current combobox inplace editor is being changed.

TComboChangeEvent=**procedure**(Sender: TObject; ACol, ARow, AltemIndex: **Integer**; ASelection: **string**) of object;

property OnComboChange: [TComboChangeEvent](#); **Description**

ACol and ARow indicate the cell in which the combobox is displayed. altemIndex indicates the new combobox selection and aSelection returns the new selection as string.

TAdvStringGrid.OnComboObjectChange

[TAdvStringGrid](#)

TComboObjectChangeEvent=**procedure**(Sender: TObject; ACol, ARow, AltRowIndex: **Integer**; ASelection: **string**; AObject: TObject) **of object**;

property OnComboObjectChange: [TComboObjectChangeEvent](#); **Description**

ACol and ARow indicate the cell in which the combobox is displayed. AltRowIndex indicates the new combobox selection and ASelection returns the new selection as string. In addition to the OnComboChange event, the Object that was added to the combobox item is also returned in this event.

TAdvStringGrid.OnContractNode

[TAdvStringGrid](#)

Event handler called when user clicks a node

TNodeClickEvent=**procedure**(Sender: TObject; ARow, ARowreal: **Integer**) of object;

property OnContractNode: [TNodeClickEvent](#); **Description**

ARow contains the displayed row index of the row where the node was clicked while ARowReal indicates the real row index, based on actual total rows

TAdvStringGrid.OnControlClick

[TAdvStringGrid](#)

TCellControlEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; CtrlID, CtrlType, CtrlVal: **string**) of **object**; **property** OnControlClick: [TCellControlEvent](#);

Description

Event triggered when an embedded HTML cell control is clicked.

CtrlID = ID of the control clicked

CtrlType = BUTTON, CHECK, COMBO, RADIO, EDIT

CtrlVal = Value of control

Control value for following control types:

CHECK = 'TRUE' or 'FALSE'

EDIT = edit text value

BUTTON = button caption

COMBO = selected combobox item

RADIO = 'TRUE' or 'FALSE'

string; Values: TStringList; var Edit: Boolean; var DropCount: Integer) of object;

TAdvStringGrid.OnControlComboList

[TAdvStringGrid](#)

TCellComboControlEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; CtrlID, CtrlType, CtrlVal: **string**; Values: [TStringList](#); **var** Edit: Boolean; **var** DropCount: **Integer**) **of object**;

property OnControlComboList: [TCellComboControlEvent](#);

Description

Event triggered when an embedded HTML cell combobox is clicked.

CtrlID = ID of the control clicked

CtrlType = BUTTON, CHECK, COMBO, RADIO, EDIT

CtrlVal = Value of control

Combobox values are set through the Values parameter.

Edit parameter determines if the combobox is a dropdown list or an editable combobox

DropCount sets the nr. of items visible in the combobox

TAdvStringGrid.OnControlEditDone

[TAdvStringGrid](#)

TCellControlEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; CtrlID, CtrlType, CtrlVal: **string**) of **object**; **property** OnControlEditDone: [TCellControlEvent](#);

Description

Event triggered when editing ends for an embedded HTML cell control

TAdvStringGrid.OnCustomCellDraw

[TAdvStringGrid](#)

TCustomCellDrawEvent=**procedure**(Sender: TObject; Canvas: [TCanvas](#); ACol, ARow: **Integer**; AState: [TGridDrawState](#); ARect: TRect; Printing: Boolean) **of object**;

property OnCustomCellDraw: [TCustomCellDrawEvent](#); **Description**

Event triggered to perform custom draw of a cell. When the Print parameter is true, this parameter indicates the custom draw needs to be performed for printing.

TAdvStringGrid.OnCustomCellSize

[TAdvStringGrid](#)

TCustomCellSizeEvent=**procedure**(Sender: TObject; Canvas: [TCanvas](#); ACol, ARow: **Integer**; var ASize: TPoint; Printing: Boolean) **of object**;

property OnCustomCellSize: [TCustomCellSizeEvent](#); **Description**

Event triggered to query the cell size for custom drawn cells during printing. The cell size dimensions is in logical pixels.

TAdvStringGrid.OnCustomCompare

[TAdvStringGrid](#)

Event handler called to compare two cells with custom function.

TCustomCompareEvent=**procedure**(Sender: TObject; str1, str2: **string**; var Res: **Integer**) of object;

property OnCustomCompare: [TCustomCompareEvent](#); **Description**

Allows comparing 2 strings by the application and return the result through the res parameter.

The result should be :

>0 : str1 > str2

=0 : str1 = str2

<0 : str1 < str2

This event is only called when ssCustom is specified through the event

TAdvStringGrid.OnDateSpinClick

[TAdvStringGrid](#)

Event handler called when the value of the current date spineditor inplace editor is being changed.

TDateTimeSpinClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; AValue: TDateTime; UpDown: Boolean) **of object**;

property OnDateSpinClick: [TDateTimeSpinClickEvent](#); **Description**

aCol and aRow indicate the cell in which the spin editor is displayed. aValue holds the new value and the updown parameter indicates whether the up or down button was pressed.

TAdvStringGrid.OnDbClickCell

[TAdvStringGrid](#)

Event handler called when left mouse is doubleclicked on a cell which returns column and row of the selected cell.

TDbClickCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**) **of object**;

property OnDbClickCell: [TDbClickCellEvent](#); **Description**

Arow and Acol indicated the double-clicked cell.

TAdvStringGrid.OnEditingDone

[TAdvStringGrid](#)

property OnEditingDone: [TNotifyEvent](#); **Description**
Event triggered when inplace editing ends

TAdvStringGrid.OnEllipsClick

[TAdvStringGrid](#)

Event handler called when user clicked the button associated with the EditBtn inplace editor or the simple button inplace editor

TEllipsClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; var S: **string**) of object;

property OnEllipsClick: [TEllipsClickEvent](#); **Description**

ACol, Arow contains the cell index where the button was clicked. The S parameter contains the cell text and can be modified by the handler.

TAdvStringGrid.OnEndColumnSize

[TAdvStringGrid](#)

Event handler is called when user stops column sizing

TEndColumnSizeEvent=**procedure**(Sender: TObject; ACol: **Integer**) of object;

property OnEndColumnSize: [TEndColumnSizeEvent](#); **Description**

ACol indicates for which the column sizing has ended.

TAdvStringGrid.OnEndRowSize

[TAdvStringGrid](#)

TEndRowSizeEvent=**procedure**(Sender: TObject; ARow: **Integer**) **of object**;

property OnEndRowSize: [TEndRowSizeEvent](#); **Description**

Event triggered after a row resize has stopped.

TAdvStringGrid.OnExpandNode

[TAdvStringGrid](#)

Event handler called when user clicks a node

TNodeClickEvent=**procedure**(Sender: TObject; ARow, ARowreal: **Integer**) of object;

property OnExpandNode: [TNodeClickEvent](#); **Description**

aRow contains the displayed row index of the row where the node was clicked while aRowReal indicates the real row index, based on actual total rows

TAdvStringGrid.OnFileProgress

[TAdvStringGrid](#)

TGridProgressEvent=**procedure**(Sender: TObject; progress: **smallint**) of object;

property OnFileProgress: [TGridProgressEvent](#); **Description**

Event called during LoadFromFile, LoadFromCSV, SaveToFile, SaveToCSV methods to indicate the progress of the operation. The progress parameter is a value between 0 and 100.

TAdvStringGrid.OnFilterProgress

[TAdvStringGrid](#)

TGridProgressEvent=**procedure**(Sender: TObject; progress: **smallint**) **of object**;

property OnFilterProgress: [TGridProgressEvent](#); **Description**

Event indicates the progress of a filtering operation. The progress value is the percentage completed.

TAdvStringGrid.OnFitToPage

[TAdvStringGrid](#)

TDoFitToPageEvent=**procedure**(Sender: TObject; **var** ScaleFactor: Double; **var** Allow: Boolean) **of** **object**;

property OnFitToPage: [TDoFitToPageEvent](#); **Description**

Set allow to false to disable fit to page, or change the scalefactor programmatically..

TAdvStringGrid.OnFloatSpinClick

[TAdvStringGrid](#)

Event handler called when the value of the current float spineditor inplace editor is being changed.

TFloatSpinClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; AValue: Double; UpDown: Boolean) **of object**;

property OnFloatSpinClick: [TFloatSpinClickEvent](#); **Description**

aCol and aRow indicate the cell in which the spin editor is displayed. aValue holds the new value and the updown parameter indicates whether the up or down button was pressed.

TAdvStringGrid.OnGetAlignment

[TAdvStringGrid](#)

Event handler called to set horizontal and vertical alignment of a cell. If no handler is specified or your handler simply returns, top left alignment is used.

TGridAlignEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; **var** HAlign: [TAlignment](#); **var** VAlign: TAlignment) **of object**;

property OnGetAlignment: [TGridAlignEvent](#); **Description**

HAlign can be taLeftJustify, taRightJustify or taCenter.

VAlign can be vtaTop, vtaCenter, vtaBottom

TAdvStringGrid.OnGetCellBorder

[TAdvStringGrid](#)

Event handler is called to set border style for each cell.

TGridBorderEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; APen: [TPen](#); **var** Borders: TCellBorders) **of object**;

property OnGetCellBorder: [TGridBorderEvent](#); **Description**

Can be used to specify what cell borders must be displayed in the grid.

TCellBorder = (cbTop,cbLeft,cbRight,cbBottom);

TCellBorders = set of TCellBorder

TAdvStringGrid.OnGetCellBorderProp

[TAdvStringGrid](#)

TGridBorderPropEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; LeftPen, TopPen, RightPen, BottomPen: [TPen](#)) **of object**;

property OnGetCellBorderProp: [TGridBorderPropEvent](#); **Description**

Event triggered to query the border pen settings for each border of a cell.

TAdvStringGrid.OnGetCellColor

[TAdvStringGrid](#)

Event handler called to set the color of a cell. If no handler is specified or your handler simply returns, default color settings are used.

TGridColorEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; AState: [TGridDrawState](#); ABrush: [TBrush](#); AFont: [TFont](#)) **of object**;

property OnGetCellColor: [TGridColorEvent](#); **Description**

ARow and ACol indicate cell for which color must be set. ABrush specifies the cell background brush. AFont can be used to modify font for a given cell.

Use the OnGetCellPrintColor to specify colors for printing or use the same event handler for both OnGetCellColor and OnGetCellPrintColor.

TAdvStringGrid.OnGetCellPrintBorder

[TAdvStringGrid](#)

Event handler is called to set print border style for each cell.

TGridBorderEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; APen: [TPen](#); **var** Borders: TCellBorders) **of object**;

property OnGetCellPrintBorder: [TGridBorderEvent](#); **Description**

Can be used to specify what cell borders must be displayed in the grid.

TCellBorder = (cbTop,cbLeft,cbRight,cbBottom);

TCellBorders = set of TCellBorder

Can be assigned to the OnGetCellBorder event handler to match on display output with printed output

TAdvStringGrid.OnGetCellPrintColor

[TAdvStringGrid](#)

Event handler called to set the color of a cell when printing. If no handler is specified or your handler simply return, default color settings are used.

TGridColorEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; AState: [TGridDrawState](#); ABrush: [TBrush](#); AFont: [TFont](#)) **of object**;

property OnGetCellPrintColor: [TGridColorEvent](#); **Description**

ARow and ACol indicate cell for which color must be set. ABrush specifies the cell background brush. AFont can be used to modify font for a given cell.

This event is identical as [OnGetCellColor](#) but for printing. Can be same event handler as OnGetCellColor or a separate event handler if needed.

TAdvStringGrid.OnGetCheckFalse

[TAdvStringGrid](#)

TGetCheckEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; var Value: **string**) **of object**;

property OnGetCheckFalse: [TGetCheckEvent](#); **Description**

Retrieves the cell value corresponding with a unchecked checkbox for the given cell

TAdvStringGrid.OnGetCheckTrue

[TAdvStringGrid](#)

TGetCheckEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; var Value: **string**) **of object**;

property OnGetCheckTrue: [TGetCheckEvent](#); **Description**

Retrieves the cell value corresponding with a checked checkbox for the given cell

TAdvStringGrid.OnGetEditorProp

[TAdvStringGrid](#)

TGetEditorPropEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; AEditLink: TEditLink) **of object**;

property OnGetEditorProp: [TGetEditorPropEvent](#); **Description**

Event called to set extra properties of custom inplace editors. The event is called after all details of the place where the control should appear are known and the window handle of the inplace editor has been created to allow further customization of the inplace edit control before editing starts.

TAdvStringGrid.OnGetEditorType

[TAdvStringGrid](#)

Event handler to query the type of inplace editor for the cell.

TGetEditorTypeEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; var AEditor: TEditorType) of **object**;

property OnGetEditorType: [TGetEditorTypeEvent](#); **Description**

For cell ACol, ARow set AEditor either one of the following editor types :

Value	Meaning
edNormal	Edit cell using edit control
edSpinEdit	Edit cell using spin edit control
edFloatSpinEdit	Edit cell using float spin edit control
edTimeSpinEdit	Edit cell using time spin edit control
edDateSpinEdit	Edit cell using date spin edit control
edComboEdit	Edit cell using default ComboBox control
edComboList	Edit cell using ComboList control
edEditBtn	Edit cell using edit box with button control
edUnitEditBtn	Edit cell using edit box + unit field control
edDateEdit	Edit cell using date validated edit control
edButton	Edit cell using button control
edNumeric	Edit cell using numeric validated edit control
edPositiveNumeric	Edit cell using positive numeric validated edit control
edFloat	Edit cell using floating point validated edit control
edCapital	Edit cell using edit control UPPER CASE only
edMixedCase	Edit cell using text validated edit control
edLowerCase	Edit cell in all lower case characters
edUpperCase	Edit cell in all upper case characters
edTimeEdit	Edit cell as time
edDateEditUpDown	Edit date with updown button
edPassword	Edit as password
edRichEdit	Rich text inplace editor
edCustom	A custom inplace is used. This is set through the <code>property</code> .

Notice that

- 1) Checkboxes are set/removed with the AddCheckbox, RemoveCheckbox procedures.
- 2) Combobox contents are set with AddComboboxString procedure or can be accessed through the ComboBox public property
- 3) The properties of the spin edit control can be set through the public SpinEdit property
- 4) The properties of the EditBtn control can be set through the public BtnEdit property.
- 5) The properties of the normal inplace edit control can be set through the NormalEdit public property

TAdvStringGrid.OnGetFormat

TAdvStringGrid

Event handler called to set datatype of cells in a column. This is called during sorts to determine method for sorting. Sorting can be alphabetic, numeric, using dates or using a custom sorting format. If custom sorting is selected, an event is called to compare cells during sorts.

TGridFormatEvent=**procedure**(Sender: TObject; ACol: **Integer**; **var** AStyle: TSortStyle; **var** aPrefix, aSuffix: **string**) **of object**;

property OnGetFormat: TGridFormatEvent; **Description**

TSortStyle can be ssAutomatic, ssAlphabetic, ssNumeric, ssDate, ssAlphaCase, ssAlphaNoCase, ssShortDateEU, ssShortDateUS, ssCustom, ssFinancial, ssAnsiAlpha, ssAnsiAlphaNoCase, ssRaw, ssHTML,ssImages

ssAutomatic : tries to determine best way of sorting

ssAlphabetic : simple alphabetic sort

ssNumeric : simple numeric sort, numbers can be integer or floating point

ssDate : relies on the standard date to string conversion for sort based on date. Depends on international settings of Windows

ssAlphaCase : case sensitive

ssAlphaNoCase : non case sensitive sort

ssShortDateEU : sort based on short european date notation dd/mm/yy

ssShortDateUS : sort based on short us date notation mm/dd/yy

ssCustom : calls the %REF% OnCustomCompare,TAdvStringGrid,OnCustomCompare %REF% event

ssRaw : calls the %REF% OnRawCompare,TAdvStringGrid,OnRawCompare %REF% event

ssFinancial : sort data with thousand and decimal separators

ssHTML : compares cell text with all HTML tags removed

ssAnsiAlpha : uses the ansi string compare functions with case sensitivity

ssAnsiAlphaNoCase : uses the ansi string compare functions without case sensitivity

Use aSuffix or aPrefix when this text must not be considered for the sort, ie when sorting \$100, \$10, ... and ignore the \$ prefix.

TAdvStringGrid.OnGridHint

[TAdvStringGrid](#)

Event handler can be used to set different hints for each cell. Set the hintstring parameter for a given column and row to specify a special hint for a given cell. Default StringGrid hint is used when no handler is defined or handler simply return.

TGridHintEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; **var** hintstr: **string**) **of object**;

property OnGridHint: [TGridHintEvent](#); **Description**

Based on the Arow, Acol parameters that indicate the cell where the hint will appear, the application can set a different hint for each cell.

TAdvStringGrid.OnIsFixedCell

[TAdvStringGrid](#)

Event handler is called to determine if a cell wherever in the grid is fixed style.

TIsFixedCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; **var** IsFixed: Boolean) **of object**;

property OnIsFixedCell: [TIsFixedCellEvent](#); **Description**

Set the IsFixed parameter to true for all fixed cells.

TAdvStringGrid.OnIsPasswordCell

[TAdvStringGrid](#)

Event handler is called to determine if a cell wherever in the grid is password style.

TIsPasswordCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; var IsPassword: Boolean) **of object**;

property OnIsPasswordCell: [TIsPasswordCellEvent](#); **Description**

Set the IsPassword parameter to true for all password cells. To allow editing as password, use the edPassword style in the OnGetEditorType event.

TAdvStringGrid.OnOleDrag

[TAdvStringGrid](#)

TOleDragDropEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; data: **string**; var Allow: Boolean) **of object**;

property OnOleDrag: [TOleDragDropEvent](#); **Description**

Event triggered before OLE drag drop starts. It indicates the column and row where the drag drop starts can start of the drag & drop operation can be disabled through the Allow parameter.

TAdvStringGrid.OnOleDragOver

[TAdvStringGrid](#)

TOleDragOverEvent=procedure(Sender: TObject; ARow, ACol: **Integer**; var Allow: Boolean) **of object**;

property OnOleDragOver: [TOleDragOverEvent](#); **Description**

Event triggered when an OLE drag & drop operation is taking place over the control. The cell where the mouse is, is returned and the drag & drop operation can be allowed or disabled through the Allow parameter.

TAdvStringGrid.OnOleDragStart

[TAdvStringGrid](#)

TOleDragStartEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**) **of object**;

property OnOleDragStart: [TOleDragStartEvent](#); **Description**

Event indicating the start of an OLE drag & drop operation from the control.

TAdvStringGrid.OnOleDragStop

[TAdvStringGrid](#)

TOleDragStopEvent=**procedure**(Sender: TObject; OLEffect: **Integer**) of object;

property OnOleDragStop: [TOleDragStopEvent](#); **Description**

Event indicating the successful termination of an OLE drag & drop operation from the control. It indicates whether it was an OLE move or copy operation.

TAdvStringGrid.OnOleDrop

[TAdvStringGrid](#)

TOleDragDropEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; data: **string**; var Allow: Boolean) **of object**;

property OnOleDrop: [TOleDragDropEvent](#); **Description**

Event triggered when an OLE drop occurs on the control. It indicates the cell where the OLE drop operation took place.

TAdvStringGrid.OnOleDropCol

[TAdvStringGrid](#)

TOleDropColEvent=**procedure**(Sender: TObject; ARow, ACol, DropCol: **Integer**) **of object**;

property OnOleDropCol: [TOleDropColEvent](#); **Description**

Event triggered when an OLE column drop occurs on the control. It indicates the cell where the OLE drop operation took place. An OLE column drop, only returns an index of the column dropped.

TAdvStringGrid.OnOleDropped

[TAdvStringGrid](#)

TOleDroppedEvent=**procedure**(Sender: TObject; ARect: [TGridRect](#)) **of object**;

property OnOleDropped: [TOleDroppedEvent](#); **Description**

Event triggered after text or a file is dropped on the grid through OLE drag&drop.

TAdvStringGrid.OnPrintNewPage

[TAdvStringGrid](#)

Event handler called after printing each row to query to start a new page.

TGridPrintNewPageEvent=**procedure**(Sender: TObject; ARow: **Integer**; **var** NewPage: Boolean) of **object**;

property OnPrintNewPage: [TGridPrintNewPageEvent](#); **Description**

The parameter is the current row being printed, and NewPage is a boolean that can be set true to start printing on a new page.

TAdvStringGrid.OnPrintPage

[TAdvStringGrid](#)

Event handler called when a page is printed (either on hardcopy or during printpreview).

TGridPrintPageEvent=**procedure**(Sender: TObject; Canvas: [TCanvas](#); PageNr, PageXSize, PageYSize: **Integer**) of object;

property OnPrintPage: [TGridPrintPageEvent](#); **Description**

This event handler creates the possibility to add custom drawing to the printing process. The Canvas is passed as a parameter and can be used to draw. Notice that changing canvas settings such as pen, font, brush affect printing other data. If a setting like a font must be changed for the custom printing process, the original settings must be restored. This can be done in following way for a font for example :

```
SaveFont := TFont.Create;  
SaveFont.Assign(Font);  
//do custom drawing here and change font  
Font.Assign(SaveFont);  
SaveFont.Free;
```

Printing is done in a canvas set to MM_LOMETRIC. This means that all units are in 1/10mm and page coordinates go from 0,0 to xsize,-ysize. There are no restrictions to the position where custom drawing is done.

TAdvStringGrid.OnPrintPageDone

[TAdvStringGrid](#)

TGridPrintPageDoneEvent=**procedure**(Sender: TObject; Canvas: [TCanvas](#); LastRow, LastRowOffset, LastPage, PageXSize, PageYSize: **Integer**) **of object**;

property OnPrintPageDone: [TGridPrintPageDoneEvent](#); **Description**

Event triggered after each page is printed.

TAdvStringGrid.OnPrintSetColumnWidth

[TAdvStringGrid](#)

Event handler called to overrule the automatically calculated column widths.

TGridPrintColumnWidthEvent=**procedure**(Sender: TObject; ACol: **Integer**; var Width: **Integer**) **of object**;

property OnPrintSetColumnWidth: [TGridPrintColumnWidthEvent](#); **Description**

Use this event handler if you need to exactly define column widths for printing. All dimensions are in 1/10mm.

TAdvStringGrid.OnPrintSetRowHeight

[TAdvStringGrid](#)

Event handler called to overrule the automatically calculated row heights.

TGridPrintRowHeightEvent=**procedure**(Sender: TObject; ARow: **Integer**; var Height: **Integer**) of object;

property OnPrintSetRowHeight: [TGridPrintRowHeightEvent](#); **Description**

Use this event handler if you need to exactly define row heights for printing. All dimensions are in 1/10mm.

TAdvStringGrid.OnPrintStart

[TAdvStringGrid](#)

Event handler called after calculation of number of pages and before printing is actually started.

TGridPrintStartEvent=**procedure**(Sender: TObject; NrofPages: **Integer**; **var** FromPage, ToPage: **Integer**) **of object**;

property OnPrintStart: [TGridPrintStartEvent](#); **Description**

This event allows to specify range of pages to be printed. NrofPages indicates the required number of pages. FromPage, ToPage are initialized to 1, NrofPages respectively. Changing the parameters FromPage, ToPage changes range of printed pages. The PrintDialog can be used to set these variables.

TAdvStringGrid.OnRadioClick

[TAdvStringGrid](#)

Event handler called when clicking a radiobutton in a radiobutton group inplace editor.

TRadioClickEvent=**procedure**(Sender: TObject; ACol, ARow, Aldx: **Integer**) **of object**;

property OnRadioClick: [TRadioClickEvent](#); **Description**

aCol and aRow indicate the cell in which the radiobutton group control is displayed. The aldX parameter is a zero based index of the selected radiobutton

TAdvStringGrid.OnRadioMouseUp

[TAdvStringGrid](#)

TRadioClickEvent=procedure(Sender: TObject; ACol, ARow, Aldx: **Integer**) **of object**;

property OnRadioMouseUp: [TRadioClickEvent](#); **Description**

Event triggered for a MouseUp on a radio button.

TAdvStringGrid.OnRawCompare

[TAdvStringGrid](#)

TRawCompareEvent=**procedure**(Sender: TObject; Col, Row1, Row2: **Integer**; var Res: **Integer**) of **object**;

property OnRawCompare: [TRawCompareEvent](#); **Description**

event handler called to compare two cells with custom function. Passes just the indexes of the cell to allow compares based on other data than just the cell text as with the OnCustomCompare event. The result of the OnRawCompare event should be set in the res parameter and according to :

>0 : cell 1 > cell 2

=0 : cell 1 = cell 2

<0 : cell 1 < cell 2

This event is only called when ssRaw is specified through the [OnGetFormat](#) event

TAdvStringGrid.OnResize

[TAdvStringGrid](#)

property OnResize: [TNotifyEvent](#); **Description**
Event triggered when the grid is resized.

TAdvStringGrid.OnRichEditSelectionChange

[TAdvStringGrid](#)

property OnRichEditSelectionChange: [TNotifyEvent](#); **Description**

Event triggered when the selection of the inplace rich text editor changes. This can be used to synchronize toolbutton properties to reflect the selection attributes of the new selected text.

TAdvStringGrid.OnRightClickCell

[TAdvStringGrid](#)

Event handler called when right mouse is pressed on a cell which returns column and row of the selected cell.

TClickCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**) of object;

property OnRightClickCell: [TClickCellEvent](#); **Description**

Arow and Acol indicated the clicked cell.

TAdvStringGrid.OnRowChanging

[TAdvStringGrid](#)

Event handler is called when the selected row is about to change

TRowChangingEvent=**procedure**(Sender: TObject; OldRow, NewRow: **Integer**; **var** Allow: Boolean) **of** **object**;

property OnRowChanging: [TRowChangingEvent](#); **Description**

Called when the grid's row changes from the OldRow to NewRow. With the Allow parameter, this change can be programmatically disabled or not.

TAdvStringGrid.OnRowMove

[TAdvStringGrid](#)

TRowSizeEvent=**procedure**(Sender: TObject; ARow: **Integer**; **var** Allow: Boolean) **of object**;

property OnRowMove: [TRowSizeEvent](#); **Description**

Event triggered during a row moving operation.

TAdvStringGrid.OnRowSize

[TAdvStringGrid](#)

TRowSizeEvent=**procedure**(Sender: TObject; ARow: **Integer**; **var** Allow: Boolean) **of object**;

property OnRowSize: [TRowSizeEvent](#); **Description**

Event triggered during a row resize operation.

TAdvStringGrid.OnScrollHint

[TAdvStringGrid](#)

Event handler is called when user is scroll-tracking to modify the scrollhint text.

TScrollHintEvent=**procedure**(Sender: TObject; ARow: **Integer**; var hintstr: **string**) **of object**;

property OnScrollHint: [TScrollHintEvent](#); **Description**

Default text is Row = ARow. Change the HintStr parameter to your custom text if required.

TAdvStringGrid.OnSpinClick

[TAdvStringGrid](#)

Event handler called when the value of the current spineditor inplace editor is being changed.

TSpinClickEvent=**procedure**(Sender: TObject; ACol, ARow, AValue: **Integer**; UpDown: Boolean) **of** **object**;

property OnSpinClick: [TSpinClickEvent](#); **Description**

ACol and ARow indicate the cell in which the spin editor is displayed. aValue holds the new value and the updown parameter indicates whether the up or down button was pressed.

TAdvStringGrid.OnTimeSpinClick

[TAdvStringGrid](#)

Event handler called when the value of the current time spineditor inplace editor is being changed.

TDateTimeSpinClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; AValue: TDateTime; UpDown: Boolean) **of object**;

property OnTimeSpinClick: [TDateTimeSpinClickEvent](#); **Description**

ACol and ARow indicate the cell in which the spin editor is displayed. aValue holds the new value and the updown parameter indicates whether the up or down button was pressed.

TAdvStringGrid.OnUpdateColumnSize

[TAdvStringGrid](#)

TUpdateColumnSizeEvent=**procedure**(Sender: TObject; ACol: **Integer**; var AWidth: **Integer**) **of object**;

property OnUpdateColumnSize: [TUpdateColumnSizeEvent](#); **Description**

Event triggered when a column size is changed. It allows to override the width set by changing the AWidth parameter.

TAdvStringGrid.AddBitButton

[TAdvStringGrid](#)

procedure AddBitButton(ACol, ARow, bw, bh: **Integer**; Caption: **string**; Glyph: [TBitmap](#); hal: [TCellHalign](#); val: [TCellValign](#)); **Description**

Adds a bitbutton to cell Acol, Arow with alignment defined by hal and val and caption by the Caption parameter. The glyph for the bitbutton is specified through the Glyph parameter. When the button is clicked, the OnButtonClick event is triggered.

TAdvStringGrid.AddBitmap

[TAdvStringGrid](#)

procedure AddBitmap(ACol, ARow: **Integer**; ABmp: [TBitmap](#); Transparent: Boolean; hal: [TCellHalign](#); val: [TCellValign](#)); **Description**

Adds a bitmap to cell Acol, Arow with alignment defined by hal and val. The transparent parameter controls display the bitmap transparently. Transparency color is determined by the pixel in the upper left corner of the bitmap.

TAdvStringGrid.AddButton

[TAdvStringGrid](#)

procedure AddButton(ACol, ARow, bw, bh: **Integer**; Caption: **string**; hal: [TCellHalign](#); val: [TCellValign](#));

Description

Adds a button to cell Acol, Arow with alignment defined by hal and val. Make sure EnableGraphics is set true.

When the button is clicked, the OnButtonClick event is triggered.

TAdvStringGrid.AddCheckBox

[TAdvStringGrid](#)

procedure AddCheckBox(ACol, ARow: **Integer**; State, Data: Boolean); **Description**

Adds a checkbox to cell Acol, Arow. The checkbox can be set checked or unchecked with the State parameter. The state of the checkbox can be dependent of the value in the cell when Data is true. In this case, the cell content takes the value of the CheckTrue property when the checkbox is checked and the value of the CheckFalse property when it is unchecked. The appearance of the checkbox is controlled by the settings in [ControlLook](#)

TAdvStringGrid.AddColumn

[TAdvStringGrid](#)

procedure AddColumn; **Description**

Adds a column to the grid.

TAdvStringGrid.AddComboString

[TAdvStringGrid](#)

procedure AddComboString(**const** s: **string**); **Description**

Add string s to the the inplace combobox editor.

TAdvStringGrid.AddComboStringObject

[TAdvStringGrid](#)

procedure AddComboStringObject(**const** s: **string**; AObject: TObject); **Description**

Adds a string and object to the inplace combobox editor. The object can be retrieved in the OnComboObjectChange event.

TAdvStringGrid.AddComment

[TAdvStringGrid](#)

procedure AddComment(ACol, ARow: **Integer**; Comment: **string**); **Description**

Adds a comment indicator to cell Acol, Arow with text comment that will popup when mouse is over the cell.

TAdvStringGrid.AddDataImage

[TAdvStringGrid](#)

procedure AddDataImage(ACol, ARow, Aidx: **Integer**; hal: [TCellHalign](#); val: [TCellValign](#)); **Description**

Adds a cell data dependent image to a cell. The alignment is equal to the alignment in AddImageIdx. Aidx specifies the index from where to start in the imagelist. Ie. if cell[acol,arow] contains 5 and Aidx=2, image 7 from the imagelist will be displayed.

TAdvStringGrid.AddFilePicture

[TAdvStringGrid](#)

procedure AddFilePicture(ACol, ARow: **Integer**; AFilePicture: TFilePicture; Transparent: Boolean; stretchmode: [TStretchMode](#); padding: **Integer**; hal: [TCellHalign](#); val: [TCellValign](#)); **Description**

Adds a picture to cell Acol, Arow with alignment defined by hal and val. The transparent parameter controls display the picture transparently. Transparency color is determined by the pixel in the upper left corner of the picture.

TAdvStringGrid.AddIcon

[TAdvStringGrid](#)

procedure AddIcon(ACol, ARow: **Integer**; AIcon: [TIcon](#); hal: [TCellHalign](#); val: [TCellValign](#)); **Description**
Adds an icon to cell Acol, Arow with alignment defined by hal and val.

TAdvStringGrid.AddImageIdx

[TAdvStringGrid](#)

procedure AddImageIdx(ACol, ARow, Aidx: **Integer**; hal: [TCellHalign](#); val: [TCellValign](#)); **Description**

Adds an image idx from the imagelist assigned to GridImages to cell Acol, Arow with alignment defined by hal and val.

hal = (haLeft, haCenter, haRight, haBeforeText, haAfterText)

val = (vaTop, vaCenter, vaBottom, vaAboveText, vaUnderText)

TAdvStringGrid.AddMarker

[TAdvStringGrid](#)

procedure AddMarker(ACol, ARow, ErrPos, ErrLen: **Integer**); **Description**

Adds an error marker line (red line under text) in the cell ACol,ARow from character position ErrPos with length of ErrLen characters.

TAdvStringGrid.AddMultiImage

[TAdvStringGrid](#)

procedure AddMultiImage(ACol, ARow, Dir: **Integer**; hal: [TCellHalign](#); val: [TCellValign](#)); **Description**
Adds a multi image cell to the grid. The indexes of the images in the image list are specified through the CellImages property. This property is an integer list, holding the indexes of the multiple images in the GridImages image list. The direction parameter is 0 for horizontal positioned multiple images and 1 for vertical positioned images.

TAdvStringGrid.AddNode

[TAdvStringGrid](#)

procedure AddNode(ARow, Span: **Integer**); **Description**

Adds a node at visible row aRow to the grid, with span indicating the number of rows it contracts or expands

TAdvStringGrid.AddPicture

[TAdvStringGrid](#)

procedure AddPicture(ACol, ARow: **Integer**; APicture: [TPicture](#); transparent: Boolean; stretchmode: [TStretchMode](#); padding: **Integer**; hal: [TCellHalign](#); val: [TCellValign](#)); **Description**

Adds a picture to cell Acol, Arow with alignment defined by hal and val. The transparent parameter controls display the picture transparently. Transparency color is determined by the pixel in the upper left corner of the picture.

With TStretchMode = (noStretch, Stretch, StretchWithAspectRatio, Shrink, ShrinkWithAspectRatio);

TAdvStringGrid.AddProgress

[TAdvStringGrid](#)

procedure AddProgress(ACol, ARow: **Integer**; FGColor, BKColor: TColor); **Description**

Adds a progress bar graphic to a cell Acol, Arow with foreground and background colors specified in the FGColor and BKColor parameters. The progressbar position reflects the cell value which should be between 0 and 100.

TAdvStringGrid.AddProgressEx

[TAdvStringGrid](#)

procedure AddProgressEx(ACol, ARow: **Integer**; FGColor, FGTextColor, BKColor, BKTextColor: TColor);

Description

Same as [AddProgress](#) but adds 2 additional color settings for the foreground and background text color of the progress bar.

TAdvStringGrid.AddRadio

[TAdvStringGrid](#)

procedure AddRadio(ACol, ARow, DirRadio, IdxRadio: **Integer**; sl: [TStrings](#)); **Description**

Adds a radiobutton to grid cell aCol,ARow. If DirRadio is true, the radiobuttons are in the vertical direction, otherwise it is in the horizontal direction. IdxRadio determines the selected radiobutton group item, while the sl stringlist holds the radiobutton labels. The appearance of the radio button is controlled by the settings in [ControlLook](#)

TAdvStringGrid.AddRotated

[TAdvStringGrid](#)

procedure AddRotated(ACol, ARow: **Integer**; AAngle: **Smallint**; s: **string**); **Description**

Adds rotated text to a cell with aAngle specified in degrees (0°-360°). The rotated text uses additionally the alignment specified through the VAlignment property and horizontal cell alignment specified through the OnGetAlignment event handler and font, color settings through the OnGetCellColor event handler.

Make sure a Truetype font is used.

TAdvStringGrid.AddRow

[TAdvStringGrid](#)

procedure AddRow; **Description**

Adds a row to the grid

TAdvStringGrid.AppendToCSV

[TAdvStringGrid](#)

procedure AppendToCSV(FileName: **String**); **Description**

Same as [SaveToCSV](#) except that this function adds grid data to the end of the file.

TAdvStringGrid.AppendToHTML

[TAdvStringGrid](#)

procedure AppendToHTML(FileName: **String**); **Description**

Same as [AppendToHTML](#) except that this function adds grid data to the end of the file.

TAdvStringGrid.AutoNumberCol

[TAdvStringGrid](#)

procedure AutoNumberCol(**const** ACol: **Integer**); **Description**

Simple autonumbering of cells in a column starting from 1 in first non-fixed row.

TAdvStringGrid.AutoNumberRow

[TAdvStringGrid](#)

procedure AutoNumberRow(**const** ARow: **Integer**); **Description**

Simple autonumbering of cells in a row starting from 1 in first non-fixed column.

TAdvStringGrid.AutoSizeCells

[TAdvStringGrid](#)

procedure AutoSizeCells(**const** DoFixedCells: Boolean; **const** PaddingX, PaddingY: **Integer**);

Description

Modifies the width and height of all cells (except fixed rows if specified) so that all text properly fits in the cells. Optionally adds extra space at right and bottom as specified in the PaddingX and PaddingY parameters.

TAdvStringGrid.AutoSizeCol

[TAdvStringGrid](#)

procedure AutoSizeCol(**const** ACol: **Integer**); **Description**

Modifies the width of column ACol so that all text properly fits in the column.

TAdvStringGrid.AutoSizeColumns

[TAdvStringGrid](#)

procedure AutoSizeColumns(**const** DoFixedCols: Boolean; **const** Padding: **Integer**); **Description**
Modifies the width of all columns (except fixed columns if specified) so that all text properly fits in the column and optionally adds extra space specified in the padding parameter.

TAdvStringGrid.AutoSizeRow

[TAdvStringGrid](#)

procedure AutoSizeRow(**const** ARow: **Integer**); **Description**

Modifies the height of row ARow so that all text properly fits in the row. Only required when using the grid in multiline mode.

TAdvStringGrid.AutoSizeRows

[TAdvStringGrid](#)

procedure AutoSizeRows(**const** DoFixedRows: Boolean; **const** Padding: Integer); **Description**

Modifies the height of all rows (except fixed rows if specified) so that all text properly fits in the rows. Only required when using the grid in multiline mode. Optionally adds extra space specified in the padding parameter.

TAdvStringGrid.BeginUpdate

[TAdvStringGrid](#)

procedure BeginUpdate; **Description**

Temporarily stops painting the grid. Can be used during lengthy update operations.

TAdvStringGrid.CalcFooter

[TAdvStringGrid](#)

```
procedure CalcFooter(ACol: Integer);
```

TAdvStringGrid.CellToReal

[TAdvStringGrid](#)

```
function CellToReal(ACol, ARow: Integer): Real;
```

TAdvStringGrid.CellToRich

[TAdvStringGrid](#)

procedure CellToRich(Col, Row: **Integer**; Richeditor: [TRichEdit](#)); **Description**

Transfers text from cell Acol, Arow to the richedit control. RichEdit can be the internal rich edit control of TAdvStringGrid which can be accessed through the RichEdit property.

TAdvStringGrid.CheckCell

[TAdvStringGrid](#)

function CheckCell(Col, Row: **Integer**): Boolean; **Description**

Use the attached CellChecker component to check the value and possibly autocorrect the value of a single cell.

TAdvStringGrid.CheckCells

[TAdvStringGrid](#)

function CheckCells(FromCol, FromRow, ToCol, ToRow: **Integer**): Boolean; **Description**

Use the attached CellChecker component to check the value and possibly autocorrect the value of a range of cells.

TAdvStringGrid.CheckCol

[TAdvStringGrid](#)

function CheckCol(DoFixed: Boolean; Col: **Integer**): Boolean; **Description**

Use the attached CellChecker component to check the value and possibly autocorrect the value of a all cells in a column

TAdvStringGrid.CheckGrid

[TAdvStringGrid](#)

function CheckGrid(DoFixed: Boolean): Boolean; **Description**

Use the attached CellChecker component to check the value and possibly autocorrect the value of a all cells in the grid.

TAdvStringGrid.CheckRow

[TAdvStringGrid](#)

function CheckRow(DoFixed: Boolean; Row: **Integer**): Boolean; **Description**

Use the attached CellChecker component to check the value and possibly autocorrect the value of a all cells in a row

TAdvStringGrid.Clear

[TAdvStringGrid](#)

procedure Clear; **Description**

Clears contents of all cells. No cells are removed, only content is cleared.

TAdvStringGrid.ClearCols

[TAdvStringGrid](#)

procedure ClearCols(CollIndex, CCount: **Integer**); **Description**

Clears contents of all cells starting from row CollIndex to CollIndex+Ccount-1.

TAdvStringGrid.ClearColumnHeaders

[TAdvStringGrid](#)

procedure ClearColumnHeaders; **Description**

Clears the content of the column headers.

TAdvStringGrid.ClearComboString

[TAdvStringGrid](#) [See also](#)

procedure ClearComboString; **Description**

Removes all strings from the inplace combobox editor

TAdvStringGrid.ClearNormalCells

[TAdvStringGrid](#)

procedure ClearNormalCells; **Description**
Clears contents of all cells except fixed cells.

TAdvStringGrid.ClearRect

[TAdvStringGrid](#)

procedure ClearRect(ACol1, ARow1, ACol2, ARow2: **Integer**); **Description**

Clears contents of all cells in rectangle indicated by [aCol1,aRow1] and [aCol2,aRow2].

TAdvStringGrid.ClearRowHeaders

[TAdvStringGrid](#)

procedure ClearRowHeaders; **Description**

Clears the content of the row headers

TAdvStringGrid.ClearRows

[TAdvStringGrid](#)

procedure ClearRows(RowIndex, RCount: **Integer**); **Description**

Clears contents of all cells starting from row RowIndex to RowIndex+Rcount-1.

TAdvStringGrid.ClearRowSelect

[TAdvStringGrid](#)

procedure ClearRowSelect; **Description**

If DisjunctRowSelect is enabled, this method clears selection of all rows.

TAdvStringGrid.ClearSelectedCells

[TAdvStringGrid](#)

procedure ClearSelectedCells; **Description**

Clears the selected cells in the grid

TAdvStringGrid.ClearSelection

[TAdvStringGrid](#)

procedure ClearSelection; **Description**

Clear the currently selected cells.

TAdvStringGrid.ColumnAvg

[TAdvStringGrid](#)

function ColumnAvg(ACol, fromRow, toRow: **Integer**): Double; **Description**

Returns the average of cell values in column ACol from row FromRow to row ToRow.

TAdvStringGrid.ColumnMax

[TAdvStringGrid](#)

function ColumnMax(ACol, fromRow, toRow: **Integer**): Double; **Description**

Returns the maximum of cell values in column ACol from row FromRow to row ToRow.

TAdvStringGrid.ColumnMin

[TAdvStringGrid](#)

function ColumnMin(ACol, fromRow, toRow: **Integer**): Double; **Description**

Returns the minimum of cell values in column ACol from row FromRow to row ToRow.

TAdvStringGrid.ColumnSum

[TAdvStringGrid](#)

function ColumnSum(ACol, fromRow, toRow: **Integer**): Double; **Description**

Returns the average of cell values in column ACol from row FromRow to row ToRow.

TAdvStringGrid.ContractAll

[TAdvStringGrid](#)

procedure ContractAll; **Description**

Contracts all nodes in the grid

TAdvStringGrid.ContractNode

[TAdvStringGrid](#)

procedure ContractNode(ARow: **Integer**); **Description**
Contracts the node at real row index aCol

TAdvStringGrid.CopySelectionToClipboard

[TAdvStringGrid](#)

procedure CopySelectionToClipboard; **Description**

Copies contents of selected cells to the clipboard.

TAdvStringGrid.CopyToClipboard

[TAdvStringGrid](#)

procedure CopyToClipboard; **Description**

Copies contents of stringgrid to the clipboard.

TAdvStringGrid.CreateBitButton

[TAdvStringGrid](#)

function CreateBitButton(ACol, ARow, bw, bh: **Integer**; Caption: **string**; hal: [TCellHalign](#); val: [TCellValign](#)): [TBitmap](#); **Description**

Creates a bitbutton in cell ACol,ARow and returns an instance of the glyph associated with the bitbutton.

TAdvStringGrid.CreateBitmap

[TAdvStringGrid](#)

function CreateBitmap(ACol, ARow: **Integer**; transparent: Boolean; hal: [TCellHalign](#); val: [TCellValign](#)): [TBitmap](#); **Description**

Creates and adds a bitmap to cell Acol, Arow with alignment defined by hal and val. The transparent parameter controls display the bitmap transparently. Transparency color is determined by the pixel in the upper left corner of the bitmap. The bitmap is automatically cleaned up when RemoveBitmap is called or the grid is destroyed.

TAdvStringGrid.CreateFilePicture

[TAdvStringGrid](#)

function CreateFilePicture(ACol, ARow: **Integer**; Transparent: Boolean; StretchMode: [TStretchMode](#); padding: **Integer**; hal: [TCellHalign](#); val: [TCellValign](#)): TFilePicture; **Description**

Creates and adds a picture to cell Acol, Arow with alignment defined by hal and val. The transparent parameter controls display the picture transparently. The picture is automatically cleaned up when RemovePicture is called or the grid is destroyed. Use TFilePicture.FileName to assign the filename.

TAdvStringGrid.Createlcon

[TAdvStringGrid](#)

function Createlcon(ACol, ARow: **Integer**; hal: [TCellHalign](#); val: [TCellValign](#)): [ticon](#); **Description**

Adds an icon to cell Acol, Arow with alignment defined by hal and val. The icon is automatically cleaned up when Removelcon is called or the grid is destroyed.

TAdvStringGrid.CreatePicture

[TAdvStringGrid](#)

function CreatePicture(ACol, ARow: **Integer**; transparent: Boolean; stretchmode: [TStretchMode](#); padding: **Integer**; hal: [TCellHalign](#); val: [TCellValign](#)): [TPicture](#); **Description**

Creates and adds a picture to cell Acol, Arow with alignment defined by hal and val. The transparent parameter controls display the picture transparently. The picture is automatically cleaned up when RemovePicture is called or the grid is destroyed. Use TPicture.LoadFromFile(filename:string) to load the image.

TAdvStringGrid.CreateRadio

[TAdvStringGrid](#)

function CreateRadio(ACol, ARow, DirRadio, IdxRadio: **Integer**): [TStrings](#); **Description**

Creates a radio button at cell ACol, ARow and returns an instance to the stringlist holding the caption text for the different radio buttons in the cell.

DirRadio is either 0 or 1 and selects vertical or horizontal direction. IdxRadio sets the index of the selected radio button. When IdxRadio is -1, the cell text is used to get and set the selected radio button.

TAdvStringGrid.CutSelectionToClipboard

[TAdvStringGrid](#)

procedure CutSelectionToClipboard; **Description**

Copies contents of the selected cells in the stringgrid to the clipboard and clears the selected cells.

TAdvStringGrid.CutToClipboard

[TAdvStringGrid](#)

procedure CutToClipboard; **Description**

Copies contents of stringgrid to the clipboard and clears all cells.

TAdvStringGrid.DisplCollIndex

[TAdvStringGrid](#)

function DisplCollIndex(ACol: **Integer**): **Integer**; **Description**

Inverse function of RealCollIndex. Returns the display column index for a given column index.

TAdvStringGrid.DisplRowIndex

[TAdvStringGrid](#)

function DisplRowIndex(ARow: **Integer**): **Integer**; **Description**

Inverse function of RealRowIndex. Returns the display column index for a given row index.

TAdvStringGrid.DoneInplaceEdit

[TAdvStringGrid](#)

procedure DoneInplaceEdit(Key: **word**; Shift: [TShiftState](#));

TAdvStringGrid.EndUpdate

[TAdvStringGrid](#)

procedure EndUpdate; **Description**

Reenables grid painting and forces a display update.

TAdvStringGrid.ExpandAll

[TAdvStringGrid](#)

procedure ExpandAll; **Description**

Expands all nodes in the grid

TAdvStringGrid.ExpandNode

[TAdvStringGrid](#)

procedure ExpandNode(ARow: **Integer**); **Description**

Expands the node at real row index aRow

TAdvStringGrid.FilterRow

[TAdvStringGrid](#)

procedure FilterRow(ARow: **Integer**);

Description

Applies the current filter set through the Filter property to a single row.

TAdvStringGrid.Find

[TAdvStringGrid](#)

function Find(StartCell: TPoint; s: **string**; FindParams: [TFindParams](#)): TPoint; **Description**
Starts a find of string s from the StartCell coordinates with the FindParams settings.

TAdvStringGrid.FindFirst

[TAdvStringGrid](#)

function FindFirst(s: **string**; FindParams: [TFindParams](#)): TPoint; **Description**

Searches the stringgrid for text s, with options for search in findparams.

This set is defined as :

TFindParameters = (fnMatchCase,fnMatchStart,fnMatchFull,fnMatchRegular,fnDirectionLeftRight,
fnFindInCurrentRow,fnFindInCurrentCol,fnIncludedFixed,fnAutoGoto)

TFindParams = set of TFindParameters;

FindFirst returns col/row of first cell matching the search specification. If no cell is found, the returned cell is (-1,-1)

For easy of use, there is a FindBusy property indicating whether the find has reached the end of the grid.

TAdvStringGrid.FindNext

[TAdvStringGrid](#)

function FindNext: TPoint; **Description**

Continues the search in the grid for text initiated by FindFirst.

For easy of use, there is a FindBusy property indicating whether the find has reached the end of the grid.

TAdvStringGrid.GetBitmap

[TAdvStringGrid](#)

function GetBitmap(ACol, ARow: **Integer**): [TBitmap](#); **Description**

Returns the bitmap object in cell Acol, Arow

TAdvStringGrid.GetCheckBoxState

[TAdvStringGrid](#)

function GetCheckBoxState(ACol, ARow: **Integer**; **var** state: Boolean): Boolean; **Description**

Gets the status of the checkbox in cell Acol, Arow and returns in result in state. If cell Acol, Arow does not contain a checkbox, the function returns false.

TAdvStringGrid.GetComboCount

[TAdvStringGrid](#)

function GetComboCount: **Integer**; **Description**

Retrieves the number of items currently in the combobox.

TAdvStringGrid.GetFilePicture

[TAdvStringGrid](#)

function GetFilePicture(ACol, ARow: **Integer**): TFilePicture; **Description**

Returns the picture object in cell Acol, Arow

TAdvStringGrid.GetImageldx

[TAdvStringGrid](#)

function GetImageldx(ACol, ARow: **Integer**; var idx: **Integer**): Boolean; **Description**

Returns true if the cell aCol, Arow contains an image from the imagelist. In the idx parameter, the actual index of the image in the imagelist is returned.

TAdvStringGrid.GetMarker

[TAdvStringGrid](#)

procedure GetMarker(ACol, ARow: **Integer**; var ErrPos, ErrLen: **Integer**); **Description**

Gets the character position and length of an error marker in the cell if it is added. When no Error Marker is added in the cell, the ErrPos returns -1.

TAdvStringGrid.GetNodeSpan

[TAdvStringGrid](#)

function GetNodeSpan(ARow: **Integer**): **Integer**; **Description**

Returns the number of rows the node is controlling

TAdvStringGrid.GetNodeSpanType

[TAdvStringGrid](#)

function GetNodeSpanType(ARow: **Integer**): **Integer**;

TAdvStringGrid.GetNodeState

[TAdvStringGrid](#)

function GetNodeState(ARow: **Integer**): Boolean; **Description**

Returns true if the node at visible row index ARow is in the expanded state

TAdvStringGrid.GetParentRow

[TAdvStringGrid](#)

function GetParentRow(ARow: **Integer**): **Integer**; **Description**

Returns the row index of the parent row for a given ARow. The parent row is the row that has the node for the row span in which this row ARow is included.

TAdvStringGrid.GetPicture

[TAdvStringGrid](#)

function GetPicture(ACol, ARow: **Integer**): [TPicture](#); **Description**

Returns the picture object in cell Acol, Arow

TAdvStringGrid.GetRadiIdx

[TAdvStringGrid](#)

function GetRadiIdx(ACol, ARow: **Integer**; var IdxRadio: **Integer**): Boolean; **Description**
Gets the index of the selected radiobutton in the radiobuttongroup at cell ACol, ARow

TAdvStringGrid.GetRadioStrings

[TAdvStringGrid](#)

function GetRadioStrings(ACol, ARow: **Integer**): [TStrings](#); **Description**

Returns the stringlist of all radio button captions in the cell ACol, ARow

TAdvStringGrid.GetRealCol

[TAdvStringGrid](#)

function GetRealCol: **Integer**; **Description**

Gets the real column index of the currently selected column

TAdvStringGrid.GetRealRow

[TAdvStringGrid](#)

function GetRealRow: **Integer**; **Description**

Gets the real row index of the currently selected row

TAdvStringGrid.GetVersionNr

[TAdvStringGrid](#)

function GetVersionNr: **Integer**; **Description**

Returns the version number of the component as integer. Low 16bit contain the minor version nr, high 16bit contain the major version nr.

TAdvStringGrid.GetVersionString

[TAdvStringGrid](#)

function GetVersionString: **string**; **Description**

Returns the version number of the component as string.

TAdvStringGrid.Group

[TAdvStringGrid](#)

procedure Group(Colindex: **Integer**); **Description**

Groups the data in the grid by column ColIndex. It creates expand / contract nodes in column 0. Is equivalent to the use of the GroupColumn property.

TAdvStringGrid.GroupAvg

[TAdvStringGrid](#)

procedure GroupAvg(Colindex: **Integer**); **Description**

Calculates the average of cell values in column ACol in all groups and displays value in group row.

TAdvStringGrid.GroupCalc

[TAdvStringGrid](#)

procedure GroupCalc(Colindex, method: **Integer**);

TAdvStringGrid.GroupCount

[TAdvStringGrid](#)

procedure GroupCount(ColIndex: **Integer**); **Description**

Calculates the number of row in all groups and displays value in group top row.

TAdvStringGrid.GroupMax

[TAdvStringGrid](#)

procedure GroupMax(Colindex: **Integer**); **Description**

Calculates the maximum of cell values in column ACol in all groups and displays value in group top row.

TAdvStringGrid.GroupMin

[TAdvStringGrid](#)

procedure GroupMin(Colindex: **Integer**); **Description**

Calculates the minimum of cell values in column ACol in all groups and displays value in group top row.

TAdvStringGrid.GroupSum

[TAdvStringGrid](#)

procedure GroupSum(Colindex: **Integer**); **Description**

Calculates the sum of cell values in column ACol in all groups and displays value in group top row.

TAdvStringGrid.HasButton

[TAdvStringGrid](#)

function HasButton(ACol, ARow: **Integer**): Boolean; **Description**

Returns true when the cell contains a button.

TAdvStringGrid.HasCheckBox

[TAdvStringGrid](#)

function HasCheckBox(ACol, ARow: **Integer**): Boolean; **Description**
Returns true when the cell contains a checkbox.

TAdvStringGrid.HasDataCheckBox

[TAdvStringGrid](#)

function HasDataCheckBox(ACol, ARow: **Integer**): Boolean; **Description**
Returns true when the cell contains a data checkbox.

TAdvStringGrid.HideCellEdit

[TAdvStringGrid](#)

procedure HideCellEdit;

TAdvStringGrid.HideColumn

[TAdvStringGrid](#)

procedure HideColumn(ColIndex: **Integer**); **Description**

Hide display of column=colindex.

TAdvStringGrid.HideColumns

[TAdvStringGrid](#)

procedure HideColumns(FromCol, ToCol: **Integer**); **Description**
Hides all columns from FromCol to ToCol.

TAdvStringGrid.HideInplaceEdit

[TAdvStringGrid](#)

procedure HideInplaceEdit; **Description**

Hides the inplace editor.

TAdvStringGrid.HideRow

[TAdvStringGrid](#)

procedure HideRow(Rowindex: **Integer**); **Description**

Hide display of row=rowindex.

TAdvStringGrid.HideRows

[TAdvStringGrid](#)

procedure HideRows(FromRow, ToRow: **Integer**); **Description**

Hides all rows from FromRow to ToRow.

TAdvStringGrid.HideRowsEx

[TAdvStringGrid](#)

procedure HideRowsEx(FromRow, ToRow: **Integer**);

TAdvStringGrid.HideSelectedRows

[TAdvStringGrid](#)

procedure HideSelectedRows; **Description**

Hides the selected rows (in disjunct row select mode) from the grid.

TAdvStringGrid.HideSelection

[TAdvStringGrid](#)

procedure HideSelection; **Description**

Hides the selected cell or cells.

TAdvStringGrid.HideUnSelectedRows

[TAdvStringGrid](#)

procedure HideUnSelectedRows; **Description**

Hides all not selected rows in the grid.

TAdvStringGrid.HighlightInCell

[TAdvStringGrid](#)

procedure HighlightInCell(DoCase: Boolean; Col, Row: **Integer**; HiText: **string**); **Description**

HiLight the word HiText with or without case sensitivity depending on the value of DoCase in the cell Col,Row.

TAdvStringGrid.HighlightInCol

[TAdvStringGrid](#)

procedure HighlightInCol(DoFixed, DoCase: Boolean; Col: **Integer**; HiText: **string**); **Description**

HiLight the word HiText with or without case sensitivity depending on the value of DoCase in all cells of the column

TAdvStringGrid.HighlightInGrid

[TAdvStringGrid](#)

procedure HighlightInGrid(DoFixed, DoCase: Boolean; HiText: **string**); **Description**

HiLight the word HiText with or without case sensitivity depending on the value of DoCase in all cells of the grid. When DoFixed is false, highlighting is not applied in the fixed cells.

TAdvStringGrid.HighlightInRow

[TAdvStringGrid](#)

procedure HighlightInRow(DoFixed, DoCase: Boolean; Row: **Integer**; HiText: **string**); **Description**

HiLight the word HiText with or without case sensitivity depending on the value of DoCase in all cells of the row. When DoFixed is false, highlighting is not applied in the fixed cells.

TAdvStringGrid.HighlightText

[TAdvStringGrid](#)

function HighlightText(DoCase: Boolean; S, Text: **string**): **string**; **Description**
HiLight the word Text in a string S.

TAdvStringGrid.InitSortXRef

[TAdvStringGrid](#) [See also](#)

procedure InitSortXRef; **Description**

Initializes the row indexing with which to access the cells in a sort indepent way.

TAdvStringGrid.InsertChildRow

[TAdvStringGrid](#)

procedure InsertChildRow(ARow: **Integer**); **Description**

Inserts a row at index ARow in the node span of a given row.

TAdvStringGrid.InsertCols

[TAdvStringGrid](#)

procedure InsertCols(CollIndex, CCount: **Integer**); **Description**
Inserts Ccount empty columns after column position CollIndex.

TAdvStringGrid.InsertFromCSV

[TAdvStringGrid](#)

procedure InsertFromCSV(FileName: **String**); **Description**

Same [LoadFromCSV](#) except that data is appended after the last row of the grid.

TAdvStringGrid.InsertRows

[TAdvStringGrid](#)

procedure InsertRows(RowIndex, RCount: **Integer**); **Description**
Inserts Rcount empty rows at row position RowIndex.

TAdvStringGrid.Invalidate

[TAdvStringGrid](#)

procedure Invalidate;

TAdvStringGrid.IsCell

[TAdvStringGrid](#)

function IsCell(SubStr: **String**; var ACol, ARow: **Integer**): Boolean;

TAdvStringGrid.IsComment

[TAdvStringGrid](#)

function IsComment(ACol, ARow: **Integer**; var comment: **string**): Boolean; **Description**
Returns true when the cell contains a comment.

TAdvStringGrid.IsHiddenColumn

[TAdvStringGrid](#)

function IsHiddenColumn(Colindex: **Integer**): Boolean; **Description**

Returns true if column=colindex is hidden.

TAdvStringGrid.IsHiddenRow

[TAdvStringGrid](#)

function IsHiddenRow(Rowindex: **Integer**): Boolean; **Description**
Returns true if row is hidden.

TAdvStringGrid.IsNode

[TAdvStringGrid](#)

function IsNode(ARow: **Integer**): Boolean; **Description**

Returns true if visible row index ARow contains a node

TAdvStringGrid.IsRadio

[TAdvStringGrid](#)

function IsRadio(ACol, ARow: **Integer**): Boolean; **Description**

Returns true of cell ACol, Arow contains a radiobutton

TAdvStringGrid.IsRotated

[TAdvStringGrid](#)

function IsRotated(ACol, ARow: **Integer**; var aAngle: **Integer**): Boolean; **Description**

If the cell contains rotation information, returns true and the angle of rotation in degrees (0°-360°)

TAdvStringGrid.IsSelected

[TAdvStringGrid](#)

function IsSelected(ACol, ARow: **Integer**): Boolean; **Description**

Returns true if the specified cell is within the selection range.

TAdvStringGrid.LoadAtPointFromBinStream

[TAdvStringGrid](#)

procedure LoadAtPointFromBinStream(Point: TPoint; Stream: [TStream](#)); **Description**
Loads cell properties from a stream starting at cell position indicated by TPoint.

TAdvStringGrid.LoadColSizes

[TAdvStringGrid](#) [See also](#)

procedure LoadColSizes; **Description**

Loads the column sizes from the registry or inifile at location pointed to by the ColumnSize property.

TAdvStringGrid.LoadFromBinFile

[TAdvStringGrid](#)

procedure LoadFromBinFile(FileName: **String**); **Description**

Loads cells from a binary file.

TAdvStringGrid.LoadFromBinStream

[TAdvStringGrid](#)

procedure LoadFromBinStream(Stream: [TStream](#)); **Description**
Loads cell information from a binary stream.

TAdvStringGrid.LoadFromCSV

[TAdvStringGrid](#) [See also](#)

procedure LoadFromCSV(FileName: **String**); **Description**

Loads contents of stringgrid from a comma delimited text file. If the [Delimiter](#) is not specified, LoadFromCSV will try to make an intelligent guess for the delimiter based on the first 2 lines of the CSV file. If Delimiter is specified, this delimiter is used regardless of what is in the file.

TAdvStringGrid.LoadFromFile

[TAdvStringGrid](#)

procedure LoadFromFile(FileName: **String**); **Description**

Loads contents as well as size of the stringgrid from a ASCII file previously created by calling the SaveToFile routine.

TAdvStringGrid.LoadFromFixed

[TAdvStringGrid](#)

procedure LoadFromFixed(FileName: **string**; positions: TIntList); **Description**

Loads the contents of stringgrid from a formatted text file. The positions parameter is a list of the fixed offsets of the columns. Create a TIntList instance, and add the positions with the Add(position:integer) method.

TAdvStringGrid.LoadFromMDBSQL

[TAdvStringGrid](#)

procedure LoadFromMDBSQL(Filename, SQL: **string**); **Description**

Loads the grid data from an Access file with the specified SQL SELECT command

TAdvStringGrid.LoadFromMDBTable

[TAdvStringGrid](#)

procedure LoadFromMDBTable(Filename, Table: **string**); **Description**

Loads the grid data from an Access file from the specified Table

TAdvStringGrid.LoadFromStream

[TAdvStringGrid](#)

procedure LoadFromStream(Stream: [TStream](#)); **Description**

Loads contents as well as size of the stringgrid from a generic stream. For example, a filestream can be used to save contents to file or a memorystream to copy data from one stringgrid to another stringgrid.

TAdvStringGrid.LoadFromXLS

[TAdvStringGrid](#)

procedure LoadFromXLS(Filename: **string**); **Description**

Uses OLE automation to transfer cell contents from XLS file with name Filename, loaded into Excel.

TAdvStringGrid.LoadFromXLSSheet

[TAdvStringGrid](#)

procedure LoadFromXLSSheet(Filename, SheetName: **string**); **Description**

Uses OLE automation to transfer cell contents from XLS file with name Filename and sheet with name SheetName in the workbook.

TAdvStringGrid.LoadPrintSettings

[TAdvStringGrid](#)

procedure LoadPrintSettings(Key, Section: **string**); **Description**
Loads the printsettings from the registry at key,section

TAdvStringGrid.MapFontHeight

[TAdvStringGrid](#)

```
function MapFontHeight(pointsize: Integer): Integer;
```

TAdvStringGrid.MapFontSize

[TAdvStringGrid](#)

```
function MapFontSize(Height: Integer): Integer;
```

TAdvStringGrid.MarkInCell

[TAdvStringGrid](#)

procedure MarkInCell(DoCase: Boolean; Col, Row: **Integer**; HiText: **string**); **Description**

Add an error marker for the word HiText with or without case sensitivity depending on the value of DoCase in the cell Col,Row.

TAdvStringGrid.MarkInCol

[TAdvStringGrid](#)

procedure MarkInCol(DoFixed, DoCase: Boolean; Col: **Integer**; HiText: **string**); **Description**

Add an error marker for the word HiText with or without case sensitivity depending on the value of DoCase in all cells of the column. When DoFixed is false, marking is not applied to the fixed cells.

TAdvStringGrid.MarkInGrid

[TAdvStringGrid](#)

procedure MarkInGrid(DoFixed, DoCase: Boolean; HiText: **string**); **Description**

Add an error marker for the word HiText with or without case sensitivity depending on the value of DoCase in all cells of the grid. When DoFixed is false, marking is not applied to the fixed cells.

TAdvStringGrid.MarkInRow

[TAdvStringGrid](#)

procedure MarkInRow(DoFixed, DoCase: Boolean; Row: **Integer**; HiText: **string**); **Description**

Add an error marker for the word HiText with or without case sensitivity depending on the value of DoCase in all cells of the row. When DoFixed is false, marking is not applied to the fixed cells.

TAdvStringGrid.MarkText

[TAdvStringGrid](#)

function MarkText(DoCase: Boolean; S, Text: **string**): **string**; **Description**
Add an error marker for the word Text in string S and returns this as a string.

TAdvStringGrid.MergeCols

[TAdvStringGrid](#)

procedure MergeCols(ColIndex1, ColIndex2: **Integer**); **Description**

Appends strings from column 2 to strings of column 1 and removes column 2.

TAdvStringGrid.MergeColumnCells

[TAdvStringGrid](#)

procedure MergeColumnCells(ColIndex: **Integer**; MainMerge: Boolean); **Description**
Automatically merges all cells with identical content in the specified column.

TAdvStringGrid.MergeRowCells

[TAdvStringGrid](#)

procedure MergeRowCells(RowIndex: **Integer**; MainMerge: Boolean); **Description**
Automatically merges all cells with identical content in the specified row.

TAdvStringGrid.MoveColumn

[TAdvStringGrid](#)

procedure MoveColumn(FromIndex, ToIndex: **Integer**); **Description**
Move contents of all cells of column FromIndex to column ToIndex.

TAdvStringGrid.MoveRow

[TAdvStringGrid](#)

procedure MoveRow(FromIndex, ToIndex: **Integer**); **Description**

Move contents of all cells of row FromIndex to row ToIndex.

TAdvStringGrid.NumHiddenColumns

[TAdvStringGrid](#)

function NumHiddenColumns: **Integer**; **Description**

Returns the number of hidden columns.

TAdvStringGrid.NumHiddenRows

[TAdvStringGrid](#)

function NumHiddenRows: **Integer**; **Description**

Returns the number of hidden rows.

TAdvStringGrid.PasteFromClipboard

[TAdvStringGrid](#)

procedure PasteFromClipboard; **Description**

Pastes data from clipboard into cells.

TAdvStringGrid.PasteSelectionFromClipboard

[TAdvStringGrid](#)

procedure PasteSelectionFromClipboard; **Description**

Pastes data from clipboard into cells at current selected cell at position row,col.

TAdvStringGrid.Print

[TAdvStringGrid](#)

procedure Print; **Description**

Prints the entire stringgrid to the selected printer, using the printer setting properties of TAdvStringGrid.

TAdvStringGrid.PrintDraw

[TAdvStringGrid](#)

procedure PrintDraw(Canvas: [TCanvas](#); DrawRect: TRect); **Description**
Draws the grid on the canvas, within the rectangle specified.

TAdvStringGrid.PrintDrawRect

[TAdvStringGrid](#)

procedure PrintDrawRect(Canvas: [TCanvas](#); DrawRect: TRect; Gridrect: [TGridRect](#)); **Description**
Draws the cells of the grid with the gridrect rectangle on the canvas, within the rectangle specified.

TAdvStringGrid.PrintPreview

[TAdvStringGrid](#)

procedure PrintPreview(Canvas: [TCanvas](#); Displayrect: TRect); **Description**

Shows a preview of the printout with exactly the same printer setting properties in the selected canvas in the rectangle defined by displayrect.

TAdvStringGrid.PrintPreviewRect

[TAdvStringGrid](#)

procedure PrintPreviewRect(Canvas: [TCanvas](#); Displayrect: TRect; Gridrect: [TGridRect](#)); **Description**
Shows a preview of the printout with exactly the same printer setting properties in the selected canvas for the cells in gridrect.

TAdvStringGrid.PrintPreviewSelectedRows

[TAdvStringGrid](#)

procedure PrintPreviewSelectedRows(Canvas: [TCanvas](#); Displayrect: TRect); **Description**
Previews the selected rows only on a Canvas.

TAdvStringGrid.PrintRect

[TAdvStringGrid](#)

procedure PrintRect(Gridrect: [TGridRect](#)); **Description**

Prints only the cells in gridrect to the selected printer, using the printer setting properties of TAdvStringGrid.

TAdvStringGrid.PrintSelectedRows

[TAdvStringGrid](#)

procedure PrintSelectedRows; **Description**

Prints the selected rows only

TAdvStringGrid.PushButton

[TAdvStringGrid](#)

procedure PushButton(ACol, ARow: **Integer**; push: Boolean); **Description**
Simulates a push button effect on the fixed cell or inplace button control

TAdvStringGrid.QSort

[TAdvStringGrid](#)

procedure QSort; **Description**

Performs a complete quicksort of the stringgrid, starting from the column indicated by the SortColumn property.

TAdvStringGrid.QSortGroup

[TAdvStringGrid](#)

procedure QSortGroup; **Description**

Performs a complete quicksort of the stringgrid, starting from the column indicated by the SortColumn property while taking grouping into account.

TAdvStringGrid.QSortIndexed

[TAdvStringGrid](#)

procedure QSortIndexed; **Description**

Performs a quicksort of the stringgrid based on the column indexes in the SortIndexes property

TAdvStringGrid.QuickSort

[TAdvStringGrid](#)

procedure QuickSort(Col, Left, Right: **Integer**); **Description**

Perform a quicksort of all cells from row left to row right for a given column col.

TAdvStringGrid.QuickSortIndexed

[TAdvStringGrid](#)

procedure QuickSortIndexed(Left, Right: **Integer**); **Description**

Perform an indexed quicksort of all cells from row left to row right for a given column col.

TAdvStringGrid.RandomFill

[TAdvStringGrid](#)

procedure RandomFill(DoFixed: Boolean; Rnd: **Integer**); **Description**

Fills the grid with random numbers. When DoFixed is true, the fixed cells are filled with random numbers as well.

TAdvStringGrid.RealColIndex

[TAdvStringGrid](#)

function RealColIndex(ACol: **Integer**): **Integer**; **Description**

Returns the real column index of the visible column index. If the number of hidden columns is 0, the real column index is always equal to the display column index. With hidden columns, the real column index of all columns after a hidden column will be incremented with the number of hidden columns before that column.

TAdvStringGrid.RealRowIndex

[TAdvStringGrid](#)

function RealRowIndex(ARow: **Integer**): **Integer**; **Description**

Returns the real row index of the visible row index. If the number of hidden rows is 0, the real row index is always equal to the display row index. With hidden rows, the real row index of all rows after a hidden row will be incremented with the number of hidden rows before that row.

TAdvStringGrid.RegisterNotifier

[TAdvStringGrid](#)

procedure RegisterNotifier(ANotifier: [TGridChangeNotifier](#));

TAdvStringGrid.RemoveBitmap

[TAdvStringGrid](#)

procedure RemoveBitmap(ACol, ARow: **Integer**); **Description**
Removes the bitmap in cell Acol, Arow

TAdvStringGrid.RemoveButton

[TAdvStringGrid](#)

procedure RemoveButton(ACol, ARow: **Integer**); **Description**

Removes the button in cell Acol, Arow

TAdvStringGrid.RemoveCheckBox

[TAdvStringGrid](#)

procedure RemoveCheckBox(ACol, ARow: **Integer**); **Description**
Removes the checkbox in cell Acol, Arow

TAdvStringGrid.RemoveChildRow

[TAdvStringGrid](#)

procedure RemoveChildRow(ARow: **Integer**); **Description**

Removes a row from a node span.

TAdvStringGrid.RemoveCols

[TAdvStringGrid](#)

procedure RemoveCols(ColIndex, CCount: **Integer**); **Description**

Removes all columns from the stringgrid, starting from column ColIndex to column ColIndex+CCount-1.

TAdvStringGrid.RemoveComboString

[TAdvStringGrid](#)

function RemoveComboString(**const** s: **string**): Boolean; **Description**
Removes string s from the combobox item list.

TAdvStringGrid.RemoveComment

[TAdvStringGrid](#)

procedure RemoveComment(ACol, ARow: **Integer**); **Description**
Removes the comment indicator from cell Acol, Arow

TAdvStringGrid.RemoveDataImage

[TAdvStringGrid](#)

procedure RemoveDataImage(ACol, ARow: **Integer**); **Description**

Removes the image from cell aCol, Arow. Note : it is not necessary to remove the image

at the cell, before adding a new image, since AddDataImage automatically cleans up previously assigned images.

TAdvStringGrid.RemoveDuplicates

[TAdvStringGrid](#)

procedure RemoveDuplicates(ACol: **Integer**; DoCase: Boolean); **Description**

Removes all rows from the grid that have duplicate data. DoCase indicates whether the check for duplicate items is case sensitive or not.

TAdvStringGrid.RemoveFilePicture

[TAdvStringGrid](#)

procedure RemoveFilePicture(ACol, ARow: **Integer**); **Description**

Removes the file picture in cell Acol, Arow

TAdvStringGrid.RemoveIcon

[TAdvStringGrid](#)

procedure RemoveIcon(ACol, ARow: **Integer**); **Description**
Removes the icon in cell Acol, Arow

TAdvStringGrid.RemoveImageldx

[TAdvStringGrid](#)

procedure RemoveImageldx(ACol, ARow: **Integer**); **Description**

Removes the image from cell aCol, Arow. Note : it is not necessary to remove the image at the cell, before adding a new image, since AddImageldx automatically cleans up previously assigned images.

TAdvStringGrid.RemoveMarker

[TAdvStringGrid](#)

procedure RemoveMarker(ACol, ARow: **Integer**); **Description**
Remove an error marker if it exists in the cell.

TAdvStringGrid.RemoveMultiImage

[TAdvStringGrid](#)

procedure RemoveMultiImage(ACol, ARow: **Integer**); **Description**
Removes the multi image cell from the grid.

TAdvStringGrid.RemoveNode

[TAdvStringGrid](#)

procedure RemoveNode(ARow: **Integer**); **Description**

Removes the node from visible row index aRow.

TAdvStringGrid.RemovePicture

[TAdvStringGrid](#)

procedure RemovePicture(ACol, ARow: **Integer**); **Description**
Removes the picture in cell Acol, Arow

TAdvStringGrid.RemoveProgress

[TAdvStringGrid](#)

procedure RemoveProgress(ACol, ARow: **Integer**); **Description**
Removes the progressbar graphic from cell Acol, Arow.

TAdvStringGrid.RemoveRadio

[TAdvStringGrid](#)

procedure RemoveRadio(ACol, ARow: **Integer**); **Description**
Removes the radiobuttons from cell aCol, Arow

TAdvStringGrid.RemoveRotated

[TAdvStringGrid](#)

procedure RemoveRotated(ACol, ARow: **Integer**); **Description**
Removes text rotation information from the cell.

TAdvStringGrid.RemoveRows

[TAdvStringGrid](#)

procedure RemoveRows(RowIndex, RCount: **Integer**); **Description**

Removes all rows from the stringgrid, starting from row RowIndex to row RowIndex+RCount-1.

TAdvStringGrid.RemoveRowsEx

[TAdvStringGrid](#)

procedure RemoveRowsEx(RowIndex, RCount: **Integer**);

TAdvStringGrid.RemoveSelectedRows

[TAdvStringGrid](#)

procedure RemoveSelectedRows; **Description**

Removes the selected rows (in disjunct row select mode) from the grid.

TAdvStringGrid.RemoveUnSelectedRows

[TAdvStringGrid](#)

procedure RemoveUnSelectedRows; **Description**

Removes all not selected rows from grid.

TAdvStringGrid.RepaintCell

[TAdvStringGrid](#)

procedure RepaintCell(c, r: **Integer**); **Description**

Repaints the cell at grid coordinates r,c

TAdvStringGrid.RepaintCol

[TAdvStringGrid](#)

procedure RepaintCol(ACol: **Integer**); **Description**
Repaints the column ACol.

TAdvStringGrid.RepaintRect

[TAdvStringGrid](#)

procedure RepaintRect(r: TRect); **Description**
Repaints the cells within rectangle r.

TAdvStringGrid.RepaintRow

[TAdvStringGrid](#)

procedure RepaintRow(ARow: **Integer**); **Description**
Repaints the row ARow.

TAdvStringGrid.ResetUpdate

[TAdvStringGrid](#)

procedure ResetUpdate; **Description**

Resets the internal update control without forcing a repaint

TAdvStringGrid.Resize

[TAdvStringGrid](#)

procedure Resize;

TAdvStringGrid.RichToCell

[TAdvStringGrid](#)

procedure RichToCell(Col, Row: **Integer**; Richeditor: [TRichEdit](#)); **Description**

Transfers text from the richedit control to cell ACol, Arow. RichEdit can be the internal rich edit control of TAdvStringGrid which can be accessed through the RichEdit property.

TAdvStringGrid.RowAvg

[TAdvStringGrid](#)

function RowAvg(ARow, fromCol, toCol: **Integer**): Double; **Description**

Returns the average of cell values in column ARow from row FromCol to row ToCol.

TAdvStringGrid.RowMax

[TAdvStringGrid](#)

function RowMax(ARow, fromCol, toCol: **Integer**): Double; **Description**

Returns the maximum of cell values in column ARow from row FromCol to row ToCol.

TAdvStringGrid.RowMin

[TAdvStringGrid](#)

function RowMin(ARow, fromCol, toCol: **Integer**): Double; **Description**

Returns the minimum of cell values in column ARow from row FromCol to row ToCol.

TAdvStringGrid.RowSum

[TAdvStringGrid](#)

function RowSum(ARow, fromCol, toCol: **Integer**): Double; **Description**

Returns the sum of cell values in column ARow from row FromCol to row ToCol.

TAdvStringGrid.SaveColSizes

[TAdvStringGrid](#) [See also](#)

procedure SaveColSizes; **Description**

Save the current column sizes into the registry or inifile at location pointed to by the ColumnSize property.

TAdvStringGrid.SavePrintSettings

[TAdvStringGrid](#)

procedure SavePrintSettings(Key, Section: **string**); **Description**
Saves the printsettings to the registry at key,section

TAdvStringGrid.SaveRectToBinStream

[TAdvStringGrid](#)

procedure SaveRectToBinStream(Rect: TRect; Stream: [TStream](#)); **Description**
Saves the rectangle of cells in binary format to a stream.

TAdvStringGrid.SaveToASCII

[TAdvStringGrid](#)

procedure SaveToASCII(FileName: **String**); **Description**

Saves contents of stringgrid to a formatted text file. If Delimiter is not specified (ie = #0) a comma is used.

TAdvStringGrid.SaveToBinFile

[TAdvStringGrid](#)

procedure SaveToBinFile(FileName: **String**); **Description**

Saves the cell information and properties in binary format to a file.

TAdvStringGrid.SaveToBinStream

[TAdvStringGrid](#)

procedure SaveToBinStream(Stream: [TStream](#)); **Description**

Saves the cell information and properties in binary format to a stream.

TAdvStringGrid.SaveToCSV

[TAdvStringGrid](#) [See also](#)

procedure SaveToCSV(FileName: **String**); **Description**

Saves contents of stringgrid to a comma delimited text file. Use the delimiter property to save with tab, space or other separators. Use SaveFixedCells property to control saving of fixed cells.

TAdvStringGrid.SaveToDOC

[TAdvStringGrid](#)

procedure SaveToDOC(Filename: **string**); **Description**

Uses OLE automation to transfer cell contents to a MS Word DOC file with table.

TAdvStringGrid.SaveToFile

[TAdvStringGrid](#)

procedure SaveToFile(FileName: **String**); **Description**

Saves contents as well as size of the stringgrid to a ASCII file so it can be restored later by calling the LoadFromFile routine.

TAdvStringGrid.SaveToFixed

[TAdvStringGrid](#)

procedure SaveToFixed(FileName: **string**; positions: TIntList); **Description**
Saves the contents of a grid to a fixed column size ASCII text file

TAdvStringGrid.SaveToHTML

[TAdvStringGrid](#)

procedure SaveToHTML(FileName: **String**); **Description**

Saves contents of stringgrid as well as alignments in HTML format. Uses the HTMLSettings property for several formatting options

TAdvStringGrid.SaveToStream

[TAdvStringGrid](#)

procedure SaveToStream(Stream: [TStream](#)); **Description**

Saves contents as well as size of the stringgrid to a generic stream.

TAdvStringGrid.SaveToXLS

[TAdvStringGrid](#)

procedure SaveToXLS(Filename: **string**); **Description**

Uses OLE automation to transfer cell contents to Excel and save to XLS file with name Filename.

TAdvStringGrid.SaveToXLSSheet

[TAdvStringGrid](#)

procedure SaveToXLSSheet(Filename, SheetName: **string**); **Description**

Uses OLE automation to transfer cell contents to Excel and save to XLS file with name Filename in the sheet specified by the sheetname.

TAdvStringGrid.SaveToXML

[TAdvStringGrid](#)

procedure SaveToXML(FileName: **String**; ListDescr, RecordDescr: **string**; FieldDescr: [TStrings](#));

Description

Saves contents of stringgrid as XML file. ListDescr specifies the XML list descriptor name, RecordDescr the record descriptor name and FieldDescr is a stringlist holding the field names for each column of the grid. Fixed columns and rows are not saved to XML.

TAdvStringGrid.ScreenToCell

[TAdvStringGrid](#)

procedure ScreenToCell(pt: TPoint; **var** ACol, ARow: Integer); **Description**

Converts the screen coordinates to a grid cell

TAdvStringGrid.ScrollInView

[TAdvStringGrid](#)

procedure ScrollInView(ColIndex, RowIndex: **Integer**); **Description**
Scrolls the cell ColIndex, RowIndex in view

TAdvStringGrid.Search

[TAdvStringGrid](#)

function Search(s: **string**): **Integer**; **Description**

Returns the first cell containing the string s, or partially matching string s.

TAdvStringGrid.SelectCols

[TAdvStringGrid](#)

procedure SelectCols(CollIndex, CCount: **Integer**); **Description**
Selects CCount columns starting from CollIndex.

TAdvStringGrid.SelectedText

[TAdvStringGrid](#)

function SelectedText: **string**; **Description**

Retrieves the contents of the selected cells as a single string

TAdvStringGrid.SelectRange

[TAdvStringGrid](#)

procedure SelectRange(FromCol, ToCol, FromRow, ToRow: **Integer**); **Description**
Selects all cells in the rectangle specified by FromCol, ToCol, FromRow, ToRow.

TAdvStringGrid.SelectRows

[TAdvStringGrid](#)

procedure SelectRows(RowIndex, RCount: **Integer**); **Description**
Selects RCount rows starting from RowIndex.

TAdvStringGrid.SetBounds

[TAdvStringGrid](#)

procedure SetBounds(ALeft, ATop, AWidth, AHeight: **Integer**);

TAdvStringGrid.SetCheckBoxState

[TAdvStringGrid](#)

function SetCheckBoxState(ACol, ARow: **Integer**; state: Boolean): Boolean; **Description**

Sets the status of the checkbox in cell Acol, Arow. If cell Acol, Arow does not contain a checkbox, the function returns false.

TAdvStringGrid.SetComboSelection

[TAdvStringGrid](#)

procedure SetComboSelection(idx: **Integer**); **Description**

Sets the current selected item to item idx in the combobox inplace editor. Can be called during the OnGetEditorType event to set default combobox items differently for each column or even cell.

TAdvStringGrid.SetComboSelectionString

[TAdvStringGrid](#)

function SetComboSelectionString(**const** s: **string**): Boolean; **Description**

Sets the current selected item to s in the combobox inplace editor. Can be called during the OnGetEditorType event to set default combobox items differently for each column or even cell.

TAdvStringGrid.SetNodeSpan

[TAdvStringGrid](#)

procedure SetNodeSpan(ARow, Span: **Integer**); **Description**

Changes the span of a given node at row ARow.

TAdvStringGrid.SetNodeState

[TAdvStringGrid](#)

procedure SetNodeState(ARow: **Integer**; Value: Boolean); **Description**

Sets the node state at visible row index aRow to the expanded state if true, to the contracted state if false.

TAdvStringGrid.SetRadiIdx

[TAdvStringGrid](#)

function SetRadiIdx(ACol, ARow, IdxRadio: **Integer**): Boolean; **Description**

Sets the index of the selected radiobutton in the radiobuttongroup at cell ACol, ARow

TAdvStringGrid.SetRotated

[TAdvStringGrid](#)

procedure SetRotated(ACol, ARow: **Integer**; AAngle: **SmallInt**); **Description**

Sets the rotation angle of the rotated text in cell ACol, ARow

TAdvStringGrid.ShowCellEdit

[TAdvStringGrid](#)

procedure ShowCellEdit;

TAdvStringGrid.ShowColumnHeaders

[TAdvStringGrid](#)

procedure ShowColumnHeaders; **Description**
Displays the column header text again in the grid

TAdvStringGrid.ShowInplaceEdit

[TAdvStringGrid](#)

procedure ShowInplaceEdit; **Description**

Shows the inplace editor in cell [grid.col,grid.row].

TAdvStringGrid.ShowRowHeaders

[TAdvStringGrid](#)

procedure ShowRowHeaders; **Description**

Displays the row header text again in the grid

TAdvStringGrid.SortByColumn

[TAdvStringGrid](#)

procedure SortByColumn(Col: **Integer**); **Description**

Sort the column col and set SortColumn property to col.

TAdvStringGrid.SortedRowIndex

[TAdvStringGrid](#)

function SortedRowIndex(ARow: **Integer**): **Integer**; **Description**

Retrieves the row index in the sorted grid that corresponds with the original unsorted row index.

TAdvStringGrid.SortLine

[TAdvStringGrid](#)

function SortLine(Col, ARow1, ARow2: **Integer**): Boolean;

TAdvStringGrid.SortSwapRows

[TAdvStringGrid](#)

procedure SortSwapRows(ARow1, ARow2: **Integer**);

TAdvStringGrid.SplitAllCells

[TAdvStringGrid](#)

procedure SplitAllCells; **Description**

Undo all cell merging in the grid.

TAdvStringGrid.SplitColumnCells

[TAdvStringGrid](#)

procedure SplitColumnCells(ColIndex: **Integer**); **Description**
Removes all cell merging from the column

TAdvStringGrid.SplitRowCells

[TAdvStringGrid](#)

procedure SplitRowCells(RowIndex: **Integer**); **Description**

Removes all cell merging from the row

TAdvStringGrid.StretchRightColumn

[TAdvStringGrid](#)

procedure StretchRightColumn; **Description**

Adapts the width of the last column to fill the grid's client rectangle

TAdvStringGrid.SwapColumns

[TAdvStringGrid](#)

procedure SwapColumns(ACol1, ACol2: **Integer**); **Description**
Swaps contents of all cells of column 1 and column 2.

TAdvStringGrid.SwapRows

[TAdvStringGrid](#)

procedure SwapRows(ARow1, ARow2: **Integer**); **Description**

Swaps contents of all cells of row 1 and row 2.

TAdvStringGrid.TextFill

[TAdvStringGrid](#)

procedure TextFill(DoFixed: Boolean; Txt: **string**); **Description**

Fills the grid with cell text Txt. When DoFixed is true, fixed cells are filled as well.

TAdvStringGrid.ToggleCheckBox

[TAdvStringGrid](#)

function ToggleCheckBox(ACol, ARow: **Integer**): Boolean; **Description**
Changes the checked state of the checkbox at acol,arow

TAdvStringGrid.UnGroup

[TAdvStringGrid](#)

procedure UnGroup; **Description**

If grouping is active, ungroups.

TAdvStringGrid.UnHideColumn

[TAdvStringGrid](#)

procedure UnHideColumn(ColIndex: **Integer**); **Description**

Shows a hidden column=colindex again.

TAdvStringGrid.UnHideColumns

[TAdvStringGrid](#)

procedure UnHideColumns(FromCol, ToCol: **Integer**); **Description**
Shows all hidden columns from FromCol to ToCol again.

TAdvStringGrid.UnHideColumnsAll

[TAdvStringGrid](#)

procedure UnHideColumnsAll; **Description**

Unhides all hidden columns

TAdvStringGrid.UnHideRow

[TAdvStringGrid](#)

procedure UnHideRow(Rowindex: **Integer**); **Description**
Shows a hidden row=rowindex again.

TAdvStringGrid.UnHideRows

[TAdvStringGrid](#)

procedure UnHideRows(FromRow, ToRow: **Integer**); **Description**
Shows all hidden rows from FromRow to ToRow again.

TAdvStringGrid.UnHideRowsAll

[TAdvStringGrid](#)

procedure UnHideRowsAll; **Description**

Unhides all hidden rows

TAdvStringGrid.UnHideSelection

[TAdvStringGrid](#)

procedure UnHideSelection; **Description**

Unhides the selected cell or cells after HideSelection

TAdvStringGrid.UnHighlightInCell

[TAdvStringGrid](#)

procedure UnHighlightInCell(Col, Row: **Integer**); **Description**
Removes all hilights from the specified cell(s)

TAdvStringGrid.UnHighlightInCol

[TAdvStringGrid](#)

procedure UnHighlightInCol(DoFixed: Boolean; Col: **Integer**); **Description**
Removes all hilights from the specified cell(s)

TAdvStringGrid.UnHighlightInGrid

[TAdvStringGrid](#)

procedure UnHighlightInGrid(DoFixed: Boolean); **Description**
Removes all hilights from the specified cell(s)

TAdvStringGrid.UnHighlightInRow

[TAdvStringGrid](#)

procedure UnHighlightInRow(DoFixed: Boolean; Row: **Integer**); **Description**
Removes all hilights from the specified cell(s)

TAdvStringGrid.UnHighlightText

[TAdvStringGrid](#)

function UnHighlightText(S: **string**): **string**; **Description**

Removes all hilights from the specified cell(s)

TAdvStringGrid.UnMarkInCell

[TAdvStringGrid](#)

procedure UnMarkInCell(Col, Row: **Integer**); **Description**
Removes all error markers from the specified cell(s)

TAdvStringGrid.UnMarkInCol

[TAdvStringGrid](#)

procedure UnMarkInCol(DoFixed: Boolean; Col: **Integer**); **Description**
Removes all error markers from the specified cell(s)

TAdvStringGrid.UnMarkInGrid

[TAdvStringGrid](#)

procedure UnMarkInGrid(DoFixed: Boolean); **Description**
Removes all error markers from the specified cell(s)

TAdvStringGrid.UnMarkInRow

[TAdvStringGrid](#)

procedure UnMarkInRow(DoFixed: Boolean; Row: **Integer**); **Description**

Removes all error markers from the specified cell(s)

TAdvStringGrid.UnMarkText

[TAdvStringGrid](#)

function UnMarkText(S: **string**): **string**; **Description**

Removes all error markers from the specified cell(s)

TAdvStringGrid.UnRegisterNotifier

TAdvStringGrid

procedure UnRegisterNotifier(ANotifier: TGridChangeNotifier);

TAdvStringGrid.UnSortedRowIndex

[TAdvStringGrid](#)

function UnSortedRowIndex(ARow: **Integer**): **Integer**; **Description**

Retrieves the unsorted row index in the sorted grid that corresponds with the new sorted row index.

TAdvStringGrid.ValidateCell

[TAdvStringGrid](#)

function ValidateCell(**const** NewValue: **string**): Boolean; **Description**
Calls the validation event for validation of the new value.

TAdvStringGrid.Zoom

[TAdvStringGrid](#)

procedure Zoom(x: Integer); **Description**

Zoom in or out on stringgrid

TAdvStringGrid.ActiveCellColor

[TAdvStringGrid](#)

property ActiveCellColor: TColor; **Description**

Sets the background color with which the fixed row / column header cells are shown when the property ActiveCellShow is true.

TAdvStringGrid.ActiveCellFont

[TAdvStringGrid](#)

property ActiveCellFont: [TFont](#); **Description**

If ActiveCellShow is true, this font is used in the fixed row and column header for the active cell. By default, this font is bold.

TAdvStringGrid.ActiveCellShow

[TAdvStringGrid](#)

property ActiveCellShow: Boolean; **Description**

When true, the fixed row and column cell for the current active cell is displayed using the ActiveCellFont.

For example, when cell 5,3 is active, fixed cells 5,0 and 0,3 will display with ActiveCellFont.

TAdvStringGrid.Alignments

[TAdvStringGrid](#)

property Alignments[i, j: **Integer**]: [TAlignment](#); **Description**
Sets the alignment per cell.

TAdvStringGrid.AllCells

[TAdvStringGrid](#)

property AllCells[j, j: Integer]: string; **Description**

Property through which the cell contents can be accessed irrespective of hidden rows in the grid.

TAdvStringGrid.AllColWidths

[TAdvStringGrid](#)

property AllColWidths[j: Integer]: Integer; **Description**

Property through which the column widths can be accessed in a way indepent of column hiding

TAdvStringGrid.AllObjects

[TAdvStringGrid](#)

property AllObjects[i, j: **Integer**]: TObject; **Description**

Property through which the cell objects can be accessed irrespective of hidden rows in the grid.

TAdvStringGrid.AlwaysQuotes

[TAdvStringGrid](#)

property AlwaysQuotes: Boolean; **Description**

When true, the export to CSV file always uses quotes. When false, quotes are only used for empty cells or cells that contain the [Delimiter](#)

TAdvStringGrid.AlwaysValidate

[TAdvStringGrid](#)

property AlwaysValidate: Boolean; **Description**

When true, the OnCellValidate event is always triggered after editing a cell ends, also when no different text than original cell content was entered.

TAdvStringGrid.AnchorHint

[TAdvStringGrid](#)

property AnchorHint: Boolean **default** False; **Description**

When true, hints are displayed for anchors in HTML formatted text

TAdvStringGrid.ArrowColor

[TAdvStringGrid](#)

property ArrowColor: TColor; **Description**

Property controlling the color of the arrows in enhanced row / column move mode.

TAdvStringGrid.AutoNumAlign

[TAdvStringGrid](#)

property AutoNumAlign: Boolean; **Description**

Enables automatic right alignment for cell containing numeric data only

TAdvStringGrid.AutoSize

[TAdvStringGrid](#)

property AutoSize: Boolean; **Description**

Setting the Autosize property to true, resizes all columns to widths so that all text just fits.

TAdvStringGrid.BackGround

[TAdvStringGrid](#)

property BackGround: [TBackground](#); **Description**

Sets the background image of the grid

TAdvStringGrid.Bands

[TAdvStringGrid](#)

property Bands: [TBands](#); **Description**

Controls the appearance of bands in the grid.

TAdvStringGrid.BtnEdit

[TAdvStringGrid](#)

property BtnEdit: [TGridEditBtn](#); **Description**

Exposes the inplace editor with attached button control for programmatic access to ButtonCaption and Glyph. In this way, ButtonCaption as well as Glyph can be different for editing different cells. Make sure to set ButtonCaption to a null string ("") when using a Glyph for proper glyph alignment.

TAdvStringGrid.BtnUnitEdit

[TAdvStringGrid](#)

property BtnUnitEdit: [TGridUnitEditBtn](#); **Description**

Exposes the inplace editor with attached button control and unit selection for programmatic access to ButtonCaption and Glyph and unit list. In this way, ButtonCaption, Glyph and unit list can be different for editing different cells. Make sure to set ButtonCaption to a null string (") when using a Glyph for proper glyph alignment.

TAdvStringGrid.CellChecker

[TAdvStringGrid](#)

property CellChecker: TAdvStringGridCheck; **Description**

Attached component (descendant of TAdvStringGridCheck) that is used for auto-correction and / or marking cell text for errors.

TAdvStringGrid.CellGraphics

[TAdvStringGrid](#)

property CellGraphics[ACol, ARow: **Integer**]: [TCellGraphic](#); **Description**

Returns the CellGraphic object controlling the graphic element in cell ACol,ARow

TAdvStringGrid.CellGraphicSize

[TAdvStringGrid](#)

property CellGraphicSize[ACol, ARow: **Integer**]: TPoint; **Description**

Returns the size of the graphic element associated with cell ACol,ARow

TAdvStringGrid.CellImages

[TAdvStringGrid](#)

property CellImages[ACol, ARow: **Integer**]: TIntList; **Description**

This public property is the integer list of indexes for a multi image cell. Indexes can be added removed, cleared through the Add, Delete, Clear methods. To add an image to the cell, use CellImages[col,row].Add(index:integer)

TAdvStringGrid.CellNode

[TAdvStringGrid](#)

property CellNode: [TCellNode](#); **Description**

Controls the cell node appearance. It has the following subproperties :

Color : background color of the node when display mode is cnFlat

ContractGlyph : glyph to display when node is in contracted state when NodeType is cnGlyph

ExpandGlyph : glyph to display when node is in expanded state when NodeType is cnGlyph

NodeColor : node color when mode is cnFlat

NodeType : controls 4 node display modes : cnFlat, cn3D, cnGlyph, cnLeaf

ShowTree : when true, a tree is displayed for the rows within the node span.

TAdvStringGrid.CellTypes

[TAdvStringGrid](#)

property CellTypes[ACol, ARow: **Integer**]: [TCellType](#); **Description**
Returns the type of the CellGraphic in cell ACol,ARow

TAdvStringGrid.CheckFalse

[TAdvStringGrid](#)

property CheckFalse: **string**; **Description**

Used with the DataCheckBox inplace editor. Reflects the value of the cell when the checkbox is not checked.

TAdvStringGrid.CheckTrue

[TAdvStringGrid](#)

property CheckTrue: **string**; **Description**

Used with the DataCheckBox inplace editor. Reflects the value of the cell when the checkbox is checked.

TAdvStringGrid.ClearTextOnly

[TAdvStringGrid](#)

property ClearTextOnly: Boolean; **Description**

When true, calling the Clear functions clears the cell content only and not the cell properties.

TAdvStringGrid.ColCount

[TAdvStringGrid](#)

property ColCount: **Integer**; **Description**

Sets the number of columns in the grid.

TAdvStringGrid.Colors

[TAdvStringGrid](#)

property Colors[i, j: **Integer**]: TColor; **Description**

Sets the background color for the cell. The OnGetCellColor event can overrule this color setting.

TAdvStringGrid.ColumnHeaders

[TAdvStringGrid](#)

property ColumnHeaders: [TStringList](#); **Description**

Allows setting header text of columns at design time. Column headers can take HTML formatted text and to specify multiline column headers the specifier can be used to indicate line breaks.

TAdvStringGrid.ColumnSize

[TAdvStringGrid](#)

property ColumnSize: [TColumnSize](#); **Description**

Controls saving and restoring of column widths as well as behaviour of column resizing.

TAdvStringGrid.Combobox

[TAdvStringGrid](#)

property Combobox: [TGridCombo](#); **Description**

Gives you access to all the properties of the inplace combobox editor. You can set sort, dropdowncount, style through this property.

TAdvStringGrid.CommentColor

[TAdvStringGrid](#) [See also](#)

property CommentColor: TColor; **Description**
Sets the color for the comment indicator.

TAdvStringGrid.Compares

[TAdvStringGrid](#)

Compares: **Integer**; **Description**

Returns the number of compare operations done during a sort.

TAdvStringGrid.ControlLook

[TAdvStringGrid](#)

property ControlLook: [TControlLook](#); **Description**

Controls the appearance of various inplace edit controls such as checkbox and radiobutton.

TAdvStringGrid.ControlValues

[TAdvStringGrid](#)

property ControlValues[ACol, ARow: **Integer**; ID: **string**]: **string**;

TAdvStringGrid.CurrentCell

[TAdvStringGrid](#)

property CurrentCell: **string**; **Description**

Property with which the current cell can be set or retrieved.

TAdvStringGrid.Cursor

[TAdvStringGrid](#)

property Cursor: TCursor; **Description**
Sets the cursor for the grid.

TAdvStringGrid.Dates

[TAdvStringGrid](#)

property Dates[ACol, ARow: **Integer**]: TDateTime; **Description**

Returns the cell content as TDateTime. An exception is raised when cell does not contain a valid date string.

TAdvStringGrid.DateTimePicker

[TAdvStringGrid](#)

property DateTimePicker: [TDateTimePicker](#); **Description**

Gives you access to all the properties of the inplace date / time editor. Can be used to set colors, min. and max. dates etc. Look at help of TDateTimePicker for all properties.

TAdvStringGrid.DefaultRowHeight

[TAdvStringGrid](#)

property DefaultRowHeight: **Integer**; **Description**

Sets the row height for all rows in the grid

TAdvStringGrid.Delimiter

[TAdvStringGrid](#)

property Delimiter: char; **Description**

Public only property to set delimiter character to use for functions SaveToCSV / LoadFromCSV, defaults to comma. Delimiter is default #0, which indicates that the grid will try to determine in an intelligent way the column delimiter. It checks characters , ; # TAB | @ '* - + and & to find the used delimiter.

TAdvStringGrid.DragDropSettings

[TAdvStringGrid](#)

property DragDropSettings: [TDragDropSettings](#); **Description**
Controls the OLE drag & drop behaviour of the grid

TAdvStringGrid.EditLink

[TAdvStringGrid](#)

property EditLink: [TEditLink](#); **Description**

Sets the EditLink component to use when the inplace editor type is edCustom.

The TEditLink component is used to do the communication between the grid and any external edit control used within the grid.

TAdvStringGrid.EditMode

[TAdvStringGrid](#)

EditMode: Boolean; **Description**

When true, the grid is in inplace editing mode

TAdvStringGrid.EditWithTags

[TAdvStringGrid](#)

property EditWithTags: Boolean **default** False; **Description**
When true, HTML tags are preserved for inplace editing.

TAdvStringGrid.EnableBlink

[TAdvStringGrid](#)

property EnableBlink: Boolean; **Description**

When true, blinking is enabled for HTML formatted text with the <BLINK> tag.

TAdvStringGrid.EnableHTML

[TAdvStringGrid](#)

property EnableHTML: Boolean; **Description**

When false, the mini HTML engine does not process any cells that contain HTML formatting tags. Default value is true.

TAdvStringGrid.EnableWheel

[TAdvStringGrid](#)

property EnableWheel: Boolean; **Description**

When true, mousewheel messages are handled by the grid, such as scrolling, panning and zooming.

TAdvStringGrid.EnhRowColMove

[TAdvStringGrid](#)

property EnhRowColMove: Boolean; **Description**

If true, shows column or row header while dragging it into the new position.

TAdvStringGrid.EnhTextSize

[TAdvStringGrid](#)

property EnhTextSize: Boolean; **Description**

If true, text in cells is appended with ... if it doesn't fit into the cell.

TAdvStringGrid.ExcelClipboardFormat

[TAdvStringGrid](#)

property ExcelClipboardFormat: Boolean; **Description**

Make clipboard functions for multiline text fully compatible with Excel.

TAdvStringGrid.ExcelStyleDecimalSeparator

[TAdvStringGrid](#)

property ExcelStyleDecimalSeparator: Boolean **default** False; **Description**

When true, the correct international decimal separator is entered when the decimal point on the numeric keypad is pressed.

TAdvStringGrid.FastPrint

[TAdvStringGrid](#)

property FastPrint: Boolean; **Description**

When true, printing is done without page calculation.

TAdvStringGrid.Filter

[TAdvStringGrid](#)

property Filter: [TFilter](#); **Description**

Holds the filter specifier for a filtering operation. Filter is a TCollection of elements with 2 properties, Condition, which is a string and Column which is an integer. The Condition property can contain a condition in following formats :

substring filtering :

S*, S? : * or ? denote multi or single character matches

>, <, >=, <= : larger than, less than, larger or equal, less or equal (when data is numeric, comparison take place based on numeric data, when a valid date is detected, date comparison is used, otherwise it defaults to alphabetic comparisons)

=, ! : equal or not equal

& : logical AND

^ : logical OR

; : logical OR

Thus, an expression : >20 & <30 is a valid filtering expression as well as !A*

TAdvStringGrid.FilterActive

[TAdvStringGrid](#)

property FilterActive: Boolean **default** False; **Description**

Set true to start filtering based on the filter specified with the Filter property.

TAdvStringGrid.FindBusy

[TAdvStringGrid](#)

property FindBusy: Boolean; **Description**

Indicates if search has started but not all cells have been searched yet.

TAdvStringGrid.FixedAsButtons

[TAdvStringGrid](#)

property FixedAsButtons: Boolean; **Description**
When true, the fixed cells behave as buttons.

TAdvStringGrid.FixedColAlways

[TAdvStringGrid](#)

property FixedColAlways: Boolean **default** False; **Description**

When true, a fixed column is always available also if the number of columns is equal to the number of fixed columns.

TAdvStringGrid.FixedCols

[TAdvStringGrid](#)

property FixedCols: **Integer**; **Description**

Sets the number of fixed columns in the grid.

TAdvStringGrid.FixedColWidth

[TAdvStringGrid](#)

property FixedColWidth: **Integer**; **Description**

Set the width of the fixed column which can otherwise not be set during design time.

TAdvStringGrid.FixedFont

[TAdvStringGrid](#)

property FixedFont: [TFont](#); **Description**

Sets the font used for fixed cells.

TAdvStringGrid.FixedFooters

[TAdvStringGrid](#)

property FixedFooters: **Integer**; **Description**

Set number of fixed rows at bottom side of the grid

TAdvStringGrid.FixedRightCols

[TAdvStringGrid](#)

property FixedRightCols: **Integer**; **Description**

Set number of fixed columns at right side of the grid

TAdvStringGrid.FixedRowAlways

[TAdvStringGrid](#)

property FixedRowAlways: Boolean **default** False; **Description**

When true, a fixed row is always available also if the number of rows is equal to the number of fixed row.

TAdvStringGrid.FixedRowHeight

[TAdvStringGrid](#)

property FixedRowHeight: **Integer**; **Description**

Set the height of the fixed row which can otherwise not be set during design time.

TAdvStringGrid.FixedRows

[TAdvStringGrid](#)

property FixedRows: **Integer**; **Description**

Sets the number of fixed rows in the grid.

TAdvStringGrid.Flat

[TAdvStringGrid](#)

property Flat: Boolean; **Description**

When true, the grid is shown in flat style.

TAdvStringGrid.FloatFormat

[TAdvStringGrid](#)

property FloatFormat: **string**; **Description**

Float formatting string used to format the cell when a float is added through the Floats[col,row] property.

TAdvStringGrid.FloatingFooter

[TAdvStringGrid](#)

property FloatingFooter: [TFloatingFooter](#); **Description**

Controls the appearance of the floating footer that can be displayed at the bottom of the grid.

TAdvStringGrid.Floats

[TAdvStringGrid](#)

property Floats[ACol, ARow: **Integer**]: double; **Description**

Property to access the cells as float values. The result is 0 if the cell doesn't contain a valid float value. When setting the float value of a cell, the FloatFormat property is used to format the float into a string.

TAdvStringGrid.FontColors

[TAdvStringGrid](#)

property FontColors[i, j: **Integer**]: TColor; **Description**

Sets the font color for a given cell.

When FontColors[Col,Row] is clNone, the default Font color is used.

TAdvStringGrid.FontNames

[TAdvStringGrid](#)

property FontNames[i, j: **Integer**]: **string**; **Description**

Sets the font name for a given cell.

When FontNames[Col,Row] is " the default Font name is used.

TAdvStringGrid.FontSizes

[TAdvStringGrid](#)

property FontSizes[i, j]: **Integer**; **Description**

Sets the font size for a given cell.

When FontSizes[Col,Row] is 0, the default Font size is used.

TAdvStringGrid.FontStyles

[TAdvStringGrid](#)

property FontStyles[i, j: **Integer**]: [TFontStyles](#); **Description**
Sets the font style for given cell.

When FontStyles[Col,Row] is [], the default Font style is used.

TAdvStringGrid.GridImages

[TAdvStringGrid](#)

property GridImages: [TImageList](#); **Description**

Set imagelist from which bitmaps are taken to display when using SetImageldx / RemoveImageldx functions or when referring to imagelist images in HTML formatting.

TAdvStringGrid.GroupColumn

[TAdvStringGrid](#)

property GroupColumn: **Integer**; **Description**

Sets the current grouping column. To ungroup, set GroupColumn to -1. Note that column 0 is the placeholder for the grouping expand / contract nodes. The appearance of the expand / contract nodes is controlled through the CellNode property.

TAdvStringGrid.HideFocusRect

[TAdvStringGrid](#)

property HideFocusRect: Boolean **default** False; **Description**

When false, disables the painting of a focusrectangle for the current focused cell.

TAdvStringGrid.HintColor

[TAdvStringGrid](#)

property HintColor: TColor; **Description**

Sets the hint color of the grid but this does not modify color of hints on other components on the same form.

TAdvStringGrid.HintShowCells

[TAdvStringGrid](#)

property HintShowCells: Boolean; **Description**

Hints are full cell contents. (Only for Delphi 3 and higher or C++Builder 3 and higher)

TAdvStringGrid.HintShowLargeText

[TAdvStringGrid](#)

property HintShowLargeText: Boolean; **Description**

When true a hint is displayed on cells that do not fit in the cell.

TAdvStringGrid.HintShowSizing

[TAdvStringGrid](#)

property HintShowSizing: Boolean; **Description**

When true, a hint is shown while columns or rows are resized indicating the new column or row size in pixels.

TAdvStringGrid.Hovering

[TAdvStringGrid](#)

property Hovering: Boolean **default** False; **Description**

When true, the cell under the mouse cursor is automatically selected when the mouse is over it for some time.

TAdvStringGrid.HTMLHint

[TAdvStringGrid](#)

property HTMLHint: Boolean **default** False; **Description**

When true, cell hints keep HTML formatting. The hint with HTML formatting can be used by the TMS THTMLHint component.

TAdvStringGrid.HTMLSettings

[TAdvStringGrid](#)

property HTMLSettings: [THTMLSettings](#); **Description**

Controls the generation of a HTML table of the grid contents :

BorderSize : table border size in HTML

CellSpacing : table cellspacing in HTML

FooterFile : file that is optionally appended at the end of the HTML for the grid content.

HeaderFile : file that is optionally included before the grid content

PrefixTag : optional HTML tags included before the grid content HTML code

SuffixTag : optional HTML tags included after the grid content HTML code

SaveColor : when true, color information is saved in HTML

SaveFonts : when true, font information is saved in HTML

TableStyle : optional table style tags for HTML

Width : width in % of the HTML table

TAdvStringGrid.ImageCache

[TAdvStringGrid](#)

property ImageCache: THTMLPictureCache;

TAdvStringGrid.InplaceRichEdit

[TAdvStringGrid](#)

property InplaceRichEdit: [TAdvRichEdit](#); **Description**

Exposes the inplace rich editor. Through this inplace editor, the selection attributes of the text can be set or retrieved.

TAdvStringGrid.IntegralHeight

[TAdvStringGrid](#)

property IntegralHeight: Boolean; **Description**

When true, the height of the grid adapts to fit exactly all rows

TAdvStringGrid.IntegralHeigth

[TAdvStringGrid](#)

property IntegralHeigth: Boolean;

Description

When true, the height of the grid adapts to make sure always full rows are displayed.

TAdvStringGrid.IntelliPan

[TAdvStringGrid](#)

property IntelliPan: [TIntelliPan](#); **Description**

Selects the panning mode. Panning can be allowed vertical, horizontal or in both directions. Panning is started by clicking the mousewheel and moving the mouse.

TAdvStringGrid.IntelliZoom

[TAdvStringGrid](#)

property IntelliZoom: Boolean **default** True; **Description**

When true, zooming with the mouse is possible. Zooming is done with holding the Ctrl key and the mousewheel.

TAdvStringGrid.Ints

[TAdvStringGrid](#)

property Ints[ACol, ARow: **Integer**]: **Integer**; **Description**

Returns the cell content as integer. An exception is raised when cell does not contain integer data and read is performed.

TAdvStringGrid.JavaCSV

[TAdvStringGrid](#)

property JavaCSV: Boolean; **Description**

Property. When true, the CSV file is in Java compatible mode, ie. line breaks are saved as ~ characters.

TAdvStringGrid.LButFlg

[TAdvStringGrid](#)

LButFlg: Boolean;

TAdvStringGrid.LockUpdate

[TAdvStringGrid](#)

property LockUpdate: Boolean;

TAdvStringGrid.Look

[TAdvStringGrid](#)

property Look: [TGridLook](#) **default** glSoft; **Description**

Sets the look of the grid either to the standard look or a soft flat look.

TAdvStringGrid.Lookup

[TAdvStringGrid](#)

property Lookup: Boolean; **Description**

TAdvStringGrid can automatically memorize all text typed in its cells and suggest autocompletion of text already entered in the grid or programmatically added in the [LookupItems](#) stringlist property

TAdvStringGrid.LookupCaseSensitive

[TAdvStringGrid](#)

property LookupCaseSensitive: Boolean; **Description**

When true, the lookups performed are case sensitive.

TAdvStringGrid.LookupHistory

[TAdvStringGrid](#)

property LookupHistory: Boolean; **Description**

Enable automatic history list building for lookup edit control.

TAdvStringGrid.LookupItems

[TAdvStringGrid](#)

property LookupItems: [TStringList](#); **Description**

Items for auto lookup of inplace edit controls. This is a stringlist to which lookup strings can be added programmatically.

Example : AdvStringGrid.LookupItems.Add('example'); or
AdvStringGrid.LookupItems.Assign(frequenttextlist);

TAdvStringGrid.MaxColWidth

[TAdvStringGrid](#) [See also](#)

property MaxColWidth: **Integer**; **Description**

Sets the maximum column width that can be set with the AutoSizeColumns function.

TAdvStringGrid.MaxEditLength

[TAdvStringGrid](#)

property MaxEditLength: **Integer**; **Description**

Sets the maximum length in characters for inplace editing.

Can be set different for each cell, by changing the value of this property in the OnGetCellEditor event.

TAdvStringGrid.MaxRowHeight

[TAdvStringGrid](#) [See also](#)

property MaxRowHeight: **Integer**; **Description**

Sets the maximum row height that can be set with the AutoSizeRows function.

TAdvStringGrid.MergedColumns

[TAdvStringGrid](#)

property MergedColumns: TIntList; **Description**

Column index list of automatically merged columns. The first index in the list is the main merge index. To automatically merge cells in columns 1,2,3, with column 1 being the main merge column, add :

```
MergedColumns.Add(1);
```

```
MergedColumns.Add(2);
```

```
MergedColumns.Add(3);
```

When `SortingSettings.AutoColumnMerge` is true, this merge sequence can be automatically reapplied after sorting

TAdvStringGrid.MinColWidth

[TAdvStringGrid](#) [See also](#)

property MinColWidth: **Integer**; **Description**

Sets the minimum column width that can be set with the AutoSizeColumns function.

TAdvStringGrid.MinRowHeight

[TAdvStringGrid](#) [See also](#)

property MinRowHeight: **Integer**; **Description**

Sets the minimum row height that can be set with the AutoSizeRows function.

TAdvStringGrid.MouseActions

[TAdvStringGrid](#)

property MouseActions: [TMouseActions](#); **Description**

Holds all settings of enhanced mouse handling in the grid

TAdvStringGrid.Multilinecells

[TAdvStringGrid](#)

property Multilinecells: Boolean; **Description**

If true, multiline cells are supported. Multiline text can be entered in cells by using the #13 line separator. For example : 'this is a'#13'multiline text' will be displayed as two lines when enabled.

As an alternative wordwrapped style can be choosen for multiline text

TAdvStringGrid.Navigation

[TAdvStringGrid](#)

property Navigation: [TNavigation](#); **Description**

Holds all settings of enhanced keyboard navigation in the grid

TAdvStringGrid.NodeState

[TAdvStringGrid](#)

property NodeState[ARow: **Integer**]: Boolean; **Description**

Returns the node state for a row. When true, the node is contracted.

TAdvStringGrid.NormalEdit

[TAdvStringGrid](#)

property NormalEdit: [TAdvInplaceEdit](#); **Description**

Exposes the normal inplace editor. Can be used to get the selected text from the inplace editor or to set additional properties if required.

TAdvStringGrid.OemConvert

[TAdvStringGrid](#)

property OemConvert: Boolean; **Description**

When true, the LoadFromCSV, SaveToCSV functions use OEM character conversion routines to load/save to ASCII CSV file.

TAdvStringGrid.OriginalCellValue

[TAdvStringGrid](#)

property OriginalCellValue: **string**; **Description**

Holds the cell value before editing started so the content can be restored at later time if for example validation fails.

TAdvStringGrid.PasswordChar

[TAdvStringGrid](#)

property PasswordChar: char; **Description**

Determines which character is used in password cells. A password cell is defined through the event OnIsPasswordCell. Default value is *

TAdvStringGrid.PictureContainer

[TAdvStringGrid](#)

property PictureContainer: TPictureContainer; **Description**

Attached picture container that can be used to hold GIF, JPEG, BMP, WMF or ICO files for use in HTML formatted cells referencing these images by their name in the PictureContainer.

TAdvStringGrid.PreviewPage

[TAdvStringGrid](#)

property PreviewPage: **Integer**; **Description**

Sets the page number to generate for page preview. Its default value is 1. This is a public property.

TAdvStringGrid.PrevSizeX

[TAdvStringGrid](#)

PrevSizeX, PrevSizeY: **Integer**;

TAdvStringGrid.PrevSizeY

[TAdvStringGrid](#)

TAdvStringGrid.PrintColEnd

[TAdvStringGrid](#)

property PrintColEnd: **Integer**; **Description**

Property indicates the column index of the last printed column on the current printed page. Property is valid during OnPrintPage event handler.

TAdvStringGrid.PrintColOffset

[TAdvStringGrid](#)

property PrintColOffset[ACol: **Integer**]: **Integer**; **Description**

Property indicates the offset from left margin of the column ACol on the current printed page. Property is valid during OnPrintPage event handler.

TAdvStringGrid.PrintColStart

[TAdvStringGrid](#)

property PrintColStart: **Integer**; **Description**

Property indicates the column index of the first printed column on the current printed page. Property is valid during OnPrintPage event handler.

TAdvStringGrid.PrintColWidth

[TAdvStringGrid](#)

property PrintColWidth[ACol: **Integer**]: **Integer**; **Description**

Property indicates the width of column ACol on the current printed page. Property is valid during OnPrintPage event handler

TAdvStringGrid.PrinterDriverFix

[TAdvStringGrid](#)

property PrinterDriverFix: Boolean; **Description**

When true, uses a fix required for some printer drivers for printing rotated fonts.

TAdvStringGrid.PrintNrOfPages

[TAdvStringGrid](#)

property PrintNrOfPages: **Integer**; **Description**

Property indicates the number of pages required for printing during the OnPrintPage event handler.

TAdvStringGrid.PrintPageRect

[TAdvStringGrid](#)

property PrintPageRect: TRect; **Description**

Property indicates the page size of the currently printed page. Can be used in the OnPrintPage event handler to add special custom printing fitted on the page.

TAdvStringGrid.PrintPageWidth

[TAdvStringGrid](#)

property PrintPageWidth: **Integer**; **Description**

Property indicates the width of the grid on the current printed page. Property is valid during OnPrintPage event handler.

TAdvStringGrid.PrintSettings

[TAdvStringGrid](#)

property PrintSettings: [TPrintSettings](#); **Description**

Holds all printout settings

TAdvStringGrid.QuoteEmptyCells

[TAdvStringGrid](#)

property QuoteEmptyCells: Boolean; **Description**

When true, empty cells are saved with double quotes in [SaveToCSV](#) and [AppendToCSV](#) calls

TAdvStringGrid.RealCol

[TAdvStringGrid](#)

property RealCol: **Integer**; **Description**

Takes hidden columns into account to return the real column index, independent of the hidden columns.

TAdvStringGrid.RealRow

[TAdvStringGrid](#)

property RealRow: **Integer**; **Description**

Takes hidden rows into account to return the real row index, independent of the hidden rows.

TAdvStringGrid.RichEdit

[TAdvStringGrid](#)

property RichEdit: [TAdvRichEdit](#); **Description**

Exposes the internal rich edit control. Can be used to do the rich text formatting of cell with its SelAttributes/SelStart/SelLength interface.

TAdvStringGrid.RowCount

[TAdvStringGrid](#)

property RowCount: **Integer**; **Description**

Sets the number of rows in the grid.

TAdvStringGrid.RowHeaders

[TAdvStringGrid](#)

property RowHeaders: [TStringList](#); **Description**

Allows setting header text of rows at design time. Row headers can take HTML formatted text and to specify multiline column headers the specifier can be used to indicate line breaks.

TAdvStringGrid.RowIndicator

[TAdvStringGrid](#)

property RowIndicator: [TBitmap](#); **Description**

Bitmap holding a row indicator glyph show in the first fixed cell of a row to indicate the current row.

TAdvStringGrid.RowSelect

[TAdvStringGrid](#)

property RowSelect[ARow: **Integer**]: Boolean; **Description**

If goRowSelect is true in Options, the grid supports disjunct row selection through the RowSelect property. Multiple disjunct rows can be selected in the grid, either programmatically, or through clicking with ctrl key. RowSelect returns true for all current selected rows. After setting selected rows programmatically, call a repaint !

TAdvStringGrid.RowSelectCount

[TAdvStringGrid](#)

property RowSelectCount: **Integer**; **Description**

Returns the number of selected rows in disjunct row select mode.

TAdvStringGrid.SaveColCount

[TAdvStringGrid](#)

property SaveColCount: **Integer**;

TAdvStringGrid.SaveEndCol

[TAdvStringGrid](#)

property SaveEndCol: **Integer**;

TAdvStringGrid.SaveEndRow

[TAdvStringGrid](#)

property SaveEndRow: Integer;

TAdvStringGrid.SaveFixedCells

[TAdvStringGrid](#)

property SaveFixedCells: Boolean; **Description**

When true, fixed cells are also saved or loaded. When false, only normal cells are saved or loaded.

TAdvStringGrid.SaveHiddenCells

[TAdvStringGrid](#)

property SaveHiddenCells: Boolean; **Description**

When true, saves the values of the hidden columns as well during SaveToCSV.
The default value is false.

TAdvStringGrid.SaveRowCount

[TAdvStringGrid](#)

property SaveRowCount: **Integer**;

TAdvStringGrid.SaveStartCol

[TAdvStringGrid](#)

property SaveStartCol: **Integer**;

TAdvStringGrid.SaveStartRow

[TAdvStringGrid](#)

property SaveStartRow: **Integer**;

TAdvStringGrid.ScrollColor

[TAdvStringGrid](#)

property ScrollColor: TColor; **Description**

Determines the background color of the scrollbars. This property has effect only when ScrollProportional is true. The default color is clNone.

TAdvStringGrid.ScrollHints

[TAdvStringGrid](#)

property ScrollHints: [TScrollHintType](#); **Description**
Sets the type of hints used during scrolling. See

TAdvStringGrid.ScrollProportional

[TAdvStringGrid](#)

property ScrollProportional: Boolean; **Description**
Enable proportional scrollbars.

TAdvStringGrid.ScrollSynch

[TAdvStringGrid](#)

property ScrollSynch: Boolean; **Description**

When true, scrolling is synchronous with scrollbar movement. The grid scrolls during the scrollbar moving instead of scrolling only when the new scrollbar position is set.

TAdvStringGrid.ScrollType

[TAdvStringGrid](#)

property ScrollType: [TScrollType](#); **Description**

Determines the type of scrollbars used. Can be either ssNormal, ssFlat or ssEncarta. This property has effect only when ScrollProportional is true.

TAdvStringGrid.ScrollWidth

[TAdvStringGrid](#)

property ScrollWidth: **Integer**; **Description**

Determines the width of the scrollbars. This property has effect only when ScrollProportional is true.

TAdvStringGrid.SelectedCell

[TAdvStringGrid](#)

property SelectedCell[i: Integer]: [TGridCoord](#);

TAdvStringGrid.SelectedCells

[TAdvStringGrid](#)

property SelectedCells[i, j: **Integer**]: Boolean; **Description**
Sets the cell selection state per cell.

TAdvStringGrid.SelectedCellsCount

[TAdvStringGrid](#)

property SelectedCellsCount: **Integer**; **Description**

Returns the number of selected cells.

TAdvStringGrid.SelectionColor

[TAdvStringGrid](#)

property SelectionColor: TColor; **Description**

Sets the grid background color for selected cells. Default value is clHighLight

TAdvStringGrid.SelectionRectangle

[TAdvStringGrid](#) [See also](#)

property SelectionRectangle: Boolean; **Description**
Draws rectangle with thick borders round selected cells.

TAdvStringGrid.SelectionRectangleColor

[TAdvStringGrid](#) [See also](#)

property SelectionRectangleColor: TColor; **Description**

Sets the color of the selection rectangle when the property SelectionRectangle is true.

TAdvStringGrid.SelectionResizer

[TAdvStringGrid](#)

property SelectionResizer: Boolean; **Description**

When SelectionResizer is true as well as SelectionRectangle, a small rectangular block is displayed in the bottom right corner of a selection rectangle with which the selection can be resized with mouse.

If [AllowSmartClipboard](#) is true, the selection resizer allows auto data filling when the selection is resized.

TAdvStringGrid.SelectionRTFKeep

[TAdvStringGrid](#)

property SelectionRTFKeep: Boolean; **Description**

When true, keeps RTF text coloring instead of changing to highlight text color

TAdvStringGrid.SelectionTextColor

[TAdvStringGrid](#)

property SelectionTextColor: TColor; **Description**

Sets the text color of selected cells. Default value is clHighLightText

TAdvStringGrid.ShowNullDates

[TAdvStringGrid](#)

property ShowNullDates: Boolean;

TAdvStringGrid.ShowSelection

[TAdvStringGrid](#)

property ShowSelection: Boolean **default** True; **Description**

When ShowSelection is false, the selected cells are not shown when the grid has no focus.

TAdvStringGrid.SizeGrowOnly

[TAdvStringGrid](#)

property SizeGrowOnly: Boolean;

TAdvStringGrid.SizeWhileTyping

[TAdvStringGrid](#)

property SizeWhileTyping: [TSizeWhileTyping](#); **Description**

Enables column sizing (vertical and/or horizontal) while typing text.

TAdvStringGrid.SizeWithForm

[TAdvStringGrid](#)

property SizeWithForm: Boolean; **Description**

If true, the stringgrid resizes itself when the parent form also resizes.

TAdvStringGrid.SortIndexes

[TAdvStringGrid](#)

property SortIndexes: TSortIndexList; **Description**
List of column indexes that determine the sort order.

TAdvStringGrid.Sortlist

[TAdvStringGrid](#)

Sortlist: [TStringList](#);

TAdvStringGrid.SortSettings

[TAdvStringGrid](#)

property SortSettings: [TSortSettings](#); **Description**
Controls the settings for grid sorting

TAdvStringGrid.SortTime

[TAdvStringGrid](#)

SortTime: DWord; **Description**

Returns the number of milliseconds required for the sort.

TAdvStringGrid.SpinEdit

[TAdvStringGrid](#)

property SpinEdit: [TGridSpin](#); **Description**

Exposes the spineditor control for programmatic access to MinValue, MaxValue and Increment properties.

TAdvStringGrid.StrippedCells

[TAdvStringGrid](#)

property StrippedCells[i, j: **Integer**]: **string**; **Description**

The StrippedCells property returns the text of a cell with all formatting removed. This is a read-only property.

TAdvStringGrid.Swaps

[TAdvStringGrid](#)

Swaps: **Integer**; **Description**

Returns the number of row swaps performed during the sort

TAdvStringGrid.Times

[TAdvStringGrid](#)

property Times[ACol, ARow: **Integer**]: TDateTime; **Description**

Returns the cell content as TDateTime. An exception is raised when cell does not contain a valid time string.

TAdvStringGrid.TMSGradientFrom

[TAdvStringGrid](#)

property TMSGradientFrom: TColor;

TAdvStringGrid.TMSGradientTo

[TAdvStringGrid](#)

property TMSGradientTo: TColor;

TAdvStringGrid.UnSortedCells

[TAdvStringGrid](#)

property UnSortedCells[i, j: **Integer**]: **string**; **Description**

Allows access to cell content as if the cells were not moved by the sort.

It is required that InitSortXRef is called before doing the sort in order to use UnSortedCells.

TAdvStringGrid.URLColor

[TAdvStringGrid](#)

property URLColor: TColor; **Description**
Color used to display URLs.

TAdvStringGrid.URLEdit

[TAdvStringGrid](#)

property URLEdit: Boolean; **Description**

URL editing enabled when right-clicking on the cell with a URL.

TAdvStringGrid.URLFull

[TAdvStringGrid](#)

property URLFull: Boolean; **Description**

If true, displays full URL, ie including protocol prefix.

TAdvStringGrid.URLShow

[TAdvStringGrid](#)

property URLShow: Boolean; **Description**

Makes grid URL-aware and displays clickable URLs in cells.

TAdvStringGrid.UseHTMLHints

[TAdvStringGrid](#)

property UseHTMLHints: Boolean;

TAdvStringGrid.VAlignment

[TAdvStringGrid](#)

property VAlignment: [TVAlignment](#); **Description**

Sets vertical alignment of text in cells in single line mode with

VAlignment = (vtaBottom, vtaCenter, vtaTop);

TAdvStringGrid.VersionNr

[TAdvStringGrid](#)

property VersionNr: **Integer**; **Description**

Returns the version number of the component as integer value. The version number consists of major and minor version number, release number and build number.

TAdvStringGrid.VersionString

[TAdvStringGrid](#)

property VersionString: **string**; **Description**

Returns the version of the component as a string.

TAdvStringGrid.WordWrap

[TAdvStringGrid](#)

property WordWrap: Boolean; **Description**

Set normal inplace editing mode and display to automatic wordwrapping. EnhTextSize and MultilineCells property should be false if wordwrapping used.

TAdvStringGrid.XYOffset

[TAdvStringGrid](#)

property XYOffset: TPoint; **Description**

Sets the offset from the topleft corner of the cell to display the cell text. Default value is 2,2 which matches with the positioning of text within the normal inplace editor.

TAdvStringGrid Events

[TAdvStringGrid](#) [Alphabetically](#)

[Legend](#)

In TAdvStringGrid

- [OnAnchorClick](#)
- [OnAnchorEnter](#)
- [OnAnchorExit](#)
- [OnAnchorHint](#)
- [OnAutoAddRow](#)
- [OnAutoDeleteRow](#)
- [OnAutoInsertCol](#)
- [OnAutoInsertRow](#)
- [OnButtonClick](#)
- [OnCanAddRow](#)
- [OnCanDeleteRow](#)
- [OnCanEditCell](#)
- [OnCanInsertRow](#)
- [OnCanSort](#)
- [OnCellChanging](#)
- [OnCellsChanged](#)
- [OnCellValidate](#)
- [OnCheckBoxClick](#)
- [OnCheckBoxMouseUp](#)
- [OnClickCell](#)
- [OnClickSort](#)
- [OnClipboardBeforePasteCell](#)
- [OnClipboardCopy](#)
- [OnClipboardCut](#)
- [OnClipboardPaste](#)
- [OnColChanging](#)
- [OnColumnMove](#)
- [OnColumnSize](#)
- [OnComboChange](#)
- [OnComboObjectChange](#)
- [OnContractNode](#)
- [OnControlClick](#)
- [OnControlComboList](#)
- [OnControlEditDone](#)
- [OnCustomCellDraw](#)
- [OnCustomCellSize](#)
- [OnCustomCompare](#)
- [OnDateSpinClick](#)
- [OnDbClickCell](#)
- [OnEditingDone](#)
- [OnEllipsClick](#)
- [OnEndColumnSize](#)
- [OnEndRowSize](#)
- [OnExpandNode](#)
- [OnFileProgress](#)
- [OnFilterProgress](#)
- [OnFitToPage](#)
- [OnFloatSpinClick](#)
- [OnGetAlignment](#)
- [OnGetCellBorder](#)
- [OnGetCellBorderProp](#)
- [OnGetCellColor](#)
- [OnGetCellPrintBorder](#)
- [OnGetCellPrintColor](#)
- [OnGetCheckFalse](#)
- [OnGetCheckTrue](#)
- [OnGetEditorProp](#)
- [OnGetEditorType](#)

- [OnGetFormat](#)
- [OnGridHint](#)
- [OnIsFixedCell](#)
- [OnIsPasswordCell](#)
- [OnOleDrag](#)
- [OnOleDragOver](#)
- [OnOleDragStart](#)
- [OnOleDragStop](#)
- [OnOleDrop](#)
- [OnOleDropCol](#)
- [OnOleDropped](#)
- [OnPrintNewPage](#)
- [OnPrintPage](#)
- [OnPrintPageDone](#)
- [OnPrintSetColumnWidth](#)
- [OnPrintSetRowHeight](#)
- [OnPrintStart](#)
- [OnRadioClick](#)
- [OnRadioMouseUp](#)
- [OnRawCompare](#)
- [OnResize](#)
- [OnRichEditSelectionChange](#)
- [OnRightClickCell](#)
- [OnRowChanging](#)
- [OnRowMove](#)
- [OnRowSize](#)
- [OnScrollHint](#)
- [OnSpinClick](#)
- [OnTimeSpinClick](#)
- [OnUpdateColumnSize](#)

Derived from TBaseGrid

- [OnGetDisplText](#)

Derived from TDrawGrid

- [OnClick](#)
- [OnColumnMoved](#)
- [OnContextPopup](#)
- [OnDbClick](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnDrawCell](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnGetEditMask](#)
- [OnGetEditText](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnMouseWheelDown](#)
- [OnMouseWheelUp](#)
- [OnRowMoved](#)
- [OnSelectCell](#)
- [OnSetEditText](#)
- [OnStartDock](#)
- [OnStartDrag](#)
- [OnTopLeftChanged](#)

Derived from TWinControl

Derived from TControl

TAdvStringGrid Events

[TAdvStringGrid](#)

[By object](#)

[Legend](#)

- [OnAnchorClick](#)
- [OnAnchorEnter](#)
- [OnAnchorExit](#)
- [OnAnchorHint](#)
- [OnAutoAddRow](#)
- [OnAutoDeleteRow](#)
- [OnAutoInsertCol](#)
- [OnAutoInsertRow](#)
- [OnButtonClick](#)
- [OnCanAddRow](#)
- [OnCanDeleteRow](#)
- [OnCanEditCell](#)
- [OnCanInsertRow](#)
- [OnCanSort](#)
- [OnCellChanging](#)
- [OnCellsChanged](#)
- [OnCellValidate](#)
- [OnCheckBoxClick](#)
- [OnCheckBoxMouseUp](#)
- [OnClick](#)
- [OnClickCell](#)
- [OnClickSort](#)
- [OnClipboardBeforePasteCell](#)
- [OnClipboardCopy](#)
- [OnClipboardCut](#)
- [OnClipboardPaste](#)
- [OnColChanging](#)
- [OnColumnMove](#)
- [OnColumnMoved](#)
- [OnColumnSize](#)
- [OnComboChange](#)
- [OnComboObjectChange](#)
- [OnContextPopup](#)
- [OnContractNode](#)
- [OnControlClick](#)
- [OnControlComboList](#)
- [OnControlEditDone](#)
- [OnCustomCellDraw](#)
- [OnCustomCellSize](#)
- [OnCustomCompare](#)
- [OnDateSpinClick](#)
- [OnDbfClick](#)
- [OnDbfClickCell](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnDrawCell](#)
- [OnEditingDone](#)
- [OnEllipsClick](#)
- [OnEndColumnSize](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEndRowSize](#)
- [OnEnter](#)
- [OnExit](#)
- [OnExpandNode](#)
- [OnFileProgress](#)
- [OnFilterProgress](#)
- [OnFitToPage](#)

- [OnFloatSpinClick](#)
- [OnGetAlignment](#)
- [OnGetCellBorder](#)
- [OnGetCellBorderProp](#)
- [OnGetCellColor](#)
- [OnGetCellPrintBorder](#)
- [OnGetCellPrintColor](#)
- [OnGetCheckFalse](#)
- [OnGetCheckTrue](#)
- [OnGetDisplText](#)
- [OnGetEditMask](#)
- [OnGetEditorProp](#)
- [OnGetEditorType](#)
- [OnGetEditText](#)
- [OnGetFormat](#)
- [OnGridHint](#)
- [OnIsFixedCell](#)
- [OnIsPasswordCell](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnMouseWheelDown](#)
- [OnMouseWheelUp](#)
- [OnOleDrag](#)
- [OnOleDragOver](#)
- [OnOleDragStart](#)
- [OnOleDragStop](#)
- [OnOleDrop](#)
- [OnOleDropCol](#)
- [OnOleDropped](#)
- [OnPrintNewPage](#)
- [OnPrintPage](#)
- [OnPrintPageDone](#)
- [OnPrintSetColumnWidth](#)
- [OnPrintSetRowHeight](#)
- [OnPrintStart](#)
- [OnRadioClick](#)
- [OnRadioMouseUp](#)
- [OnRawCompare](#)
- [OnResize](#)
- [OnRichEditSelectionChange](#)
- [OnRightClickCell](#)
- [OnRowChanging](#)
- [OnRowMove](#)
- [OnRowMoved](#)
- [OnRowSize](#)
- [OnScrollHint](#)
- [OnSelectCell](#)
- [OnSetEditText](#)
- [OnSpinClick](#)
- [OnStartDock](#)
- [OnStartDrag](#)
- [OnTimeSpinClick](#)
- [OnTopLeftChanged](#)
- [OnUpdateColumnSize](#)

TAdvStringGrid Methods

[TAdvStringGrid](#) [Alphabetically](#) [Legend](#)

In TAdvStringGrid

[AddBitButton](#)
[AddBitmap](#)
[AddButton](#)
[AddCheckBox](#)
[AddColumn](#)
[AddComboString](#)
[AddComboStringObject](#)
[AddComment](#)
[AddDataImage](#)
[AddFilePicture](#)
[AddIcon](#)
[AddImageIdx](#)
[AddMarker](#)
[AddMultiImage](#)
[AddNode](#)
[AddPicture](#)
[AddProgress](#)
[AddProgressEx](#)
[AddRadio](#)
[AddRotated](#)
[AddRow](#)
[AppendToCSV](#)
[AppendToHTML](#)
[AutoNumberCol](#)
[AutoNumberRow](#)
[AutoSizeCells](#)
[AutoSizeCol](#)
[AutoSizeColumns](#)
[AutoSizeRow](#)
[AutoSizeRows](#)
[BeginUpdate](#)
[CalcFooter](#)
[CellToReal](#)
[CellToRich](#)
[CheckCell](#)
[CheckCells](#)
[CheckCol](#)
[CheckGrid](#)
[CheckRow](#)
[Clear](#)
[ClearCols](#)
[ClearColumnHeaders](#)
[ClearComboString](#)
[ClearNormalCells](#)
[ClearRect](#)
[ClearRowHeaders](#)
[ClearRows](#)
[ClearRowSelect](#)
[ClearSelectedCells](#)
[ClearSelection](#)
[ColumnAvg](#)
[ColumnMax](#)
[ColumnMin](#)
[ColumnSum](#)
[ContractAll](#)
[ContractNode](#)
[CopySelectionToClipboard](#)
[CopyToClipboard](#)

[CreateBitButton](#)
[CreateBitmap](#)
[CreateFilePicture](#)
[CreateIcon](#)
[CreatePicture](#)
[CreateRadio](#)
[CutSelectionToClipboard](#)
[CutToClipboard](#)
[DisplColIndex](#)
[DisplRowIndex](#)
[DoneInplaceEdit](#)
[EndUpdate](#)
[ExpandAll](#)
[ExpandNode](#)
[FilterRow](#)
[Find](#)
[FindFirst](#)
[FindNext](#)
[GetBitmap](#)
[GetCheckBoxState](#)
[GetComboCount](#)
[GetFilePicture](#)
[GetImageldx](#)
[GetMarker](#)
[GetNodeSpan](#)
[GetNodeSpanType](#)
[GetNodeState](#)
[GetParentRow](#)
[GetPicture](#)
[GetRadioldx](#)
[GetRadioStrings](#)
[GetRealCol](#)
[GetRealRow](#)
[GetVersionNr](#)
[GetVersionString](#)
[Group](#)
[GroupAvg](#)
[GroupCalc](#)
[GroupCount](#)
[GroupMax](#)
[GroupMin](#)
[GroupSum](#)
[HasButton](#)
[HasCheckBox](#)
[HasDataCheckBox](#)
[HideCellEdit](#)
[HideColumn](#)
[HideColumns](#)
[HideInplaceEdit](#)
[HideRow](#)
[HideRows](#)
[HideRowsEx](#)
[HideSelectedRows](#)
[HideSelection](#)
[HideUnSelectedRows](#)
[HilighInCell](#)
[HilighInCol](#)
[HilighInGrid](#)
[HilighInRow](#)
[HilighText](#)
[InitSortXRef](#)
[InsertChildRow](#)

[InsertCols](#)
[InsertFromCSV](#)
[InsertRows](#)
[Invalidate](#)
[IsCell](#)
[IsComment](#)
[IsHiddenColumn](#)
[IsHiddenRow](#)
[IsNode](#)
[IsRadio](#)
[IsRotated](#)
[IsSelected](#)
[LoadAtPointFromBinStream](#)
[LoadColSizes](#)
[LoadFromBinFile](#)
[LoadFromBinStream](#)
[LoadFromCSV](#)
[LoadFromFile](#)
[LoadFromFixed](#)
[LoadFromMDBSQL](#)
[LoadFromMDBTable](#)
[LoadFromStream](#)
[LoadFromXLS](#)
[LoadFromXLSSheet](#)
[LoadPrintSettings](#)
[MapFontHeight](#)
[MapFontSize](#)
[MarkInCell](#)
[MarkInCol](#)
[MarkInGrid](#)
[MarkInRow](#)
[MarkText](#)
[MergeCols](#)
[MergeColumnCells](#)
[MergeRowCells](#)
[MoveColumn](#)
[MoveRow](#)
[NumHiddenColumns](#)
[NumHiddenRows](#)
[PasteFromClipboard](#)
[PasteSelectionFromClipboard](#)
[Print](#)
[PrintDraw](#)
[PrintDrawRect](#)
[PrintPreview](#)
[PrintPreviewRect](#)
[PrintPreviewSelectedRows](#)
[PrintRect](#)
[PrintSelectedRows](#)
[PushButton](#)
[QSort](#)
[QSortGroup](#)
[QSortIndexed](#)
[QuickSort](#)
[QuickSortIndexed](#)
[RandomFill](#)
[RealColIndex](#)
[RealRowIndex](#)
[RegisterNotifier](#)
[RemoveBitmap](#)
[RemoveButton](#)
[RemoveCheckBox](#)

[RemoveChildRow](#)
[RemoveCols](#)
[RemoveComboString](#)
[RemoveComment](#)
[RemoveDataImage](#)
[RemoveDuplicates](#)
[RemoveFilePicture](#)
[RemoveIcon](#)
[RemoveImageIdx](#)
[RemoveMarker](#)
[RemoveMultiImage](#)
[RemoveNode](#)
[RemovePicture](#)
[RemoveProgress](#)
[RemoveRadio](#)
[RemoveRotated](#)
[RemoveRows](#)
[RemoveRowsEx](#)
[RemoveSelectedRows](#)
[RemoveUnSelectedRows](#)
[RepaintCell](#)
[RepaintCol](#)
[RepaintRect](#)
[RepaintRow](#)
[ResetUpdate](#)
[Resize](#)
[RichToCell](#)
[RowAvg](#)
[RowMax](#)
[RowMin](#)
[RowSum](#)
[SaveColSizes](#)
[SavePrintSettings](#)
[SaveRectToBinStream](#)
[SaveToASCII](#)
[SaveToBinFile](#)
[SaveToBinStream](#)
[SaveToCSV](#)
[SaveToDOC](#)
[SaveToFile](#)
[SaveToFixed](#)
[SaveToHTML](#)
[SaveToStream](#)
[SaveToXLS](#)
[SaveToXLSSheet](#)
[SaveToXML](#)
[ScreenToCell](#)
[ScrollInView](#)
[Search](#)
[SelectCols](#)
[SelectedText](#)
[SelectRange](#)
[SelectRows](#)
[SetBounds](#)
[SetCheckBoxState](#)
[SetComboSelection](#)
[SetComboSelectionString](#)
[SetNodeSpan](#)
[SetNodeState](#)
[SetRadiIdx](#)
[SetRotated](#)
[ShowCellEdit](#)

[ShowColumnHeaders](#)
[ShowInplaceEdit](#)
[ShowRowHeaders](#)
[SortByColumn](#)
[SortedRowIndex](#)
[SortLine](#)
[SortSwapRows](#)
[SplitAllCells](#)
[SplitColumnCells](#)
[SplitRowCells](#)
[StretchRightColumn](#)
[SwapColumns](#)
[SwapRows](#)
[TextFill](#)
[ToggleCheckBox](#)
[UnGroup](#)
[UnHideColumn](#)
[UnHideColumns](#)
[UnHideColumnsAll](#)
[UnHideRow](#)
[UnHideRows](#)
[UnHideRowsAll](#)
[UnHideSelection](#)
[UnHighlightInCell](#)
[UnHighlightInCol](#)
[UnHighlightInGrid](#)
[UnHighlightInRow](#)
[UnHighlightText](#)
[UnMarkInCell](#)
[UnMarkInCol](#)
[UnMarkInGrid](#)
[UnMarkInRow](#)
[UnMarkText](#)
[UnRegisterNotifier](#)
[UnSortedRowIndex](#)
[ValidateCell](#)
[Zoom](#)

Derived from TBaseGrid

[BaseCell](#)
[CellRect](#)
[CellSize](#)
[CellSpan](#)
[ColSpanIdentical](#)
[FullCell](#)
[IsBaseCell](#)
[IsMergedCell](#)
[IsXMergedCell](#)
[IsYMergedCell](#)
[MaxRowSpan](#)
[MergeCells](#)
[MinRowSpan](#)
[RowSpanIdentical](#)
[SplitCells](#)

Derived from TStringGrid

[Create](#)
[Destroy](#)

Derived from TCustomDrawGrid

[MouseToCell](#)

Derived from TCustomGrid

[MouseCoord](#)

Derived from TCustomControl

Derived from TWinControl

[Broadcast](#)

[CanFocus](#)

[ContainsControl](#)

[ControlAtPos](#)

[CreateParented](#)

[DefaultHandler](#)

[DisableAlign](#)

[DockDrop](#)

[EnableAlign](#)

[FindChildControl](#)

[FlipChildren](#)

[Focused](#)

[GetTabOrderList](#)

[HandleAllocated](#)

[HandleNeeded](#)

[InsertControl](#)

[PaintTo](#)

[Realign](#)

[RemoveControl](#)

[Repaint](#)

[ScaleBy](#)

[ScrollBy](#)

[SetFocus](#)

[Update](#)

[UpdateControlState](#)

Derived from TControl

[BeginDrag](#)

[BringToFront](#)

[ClientToParent](#)

[ClientToScreen](#)

[Dock](#)

[DragDrop](#)

[Dragging](#)

[DrawTextBiDiModeFlags](#)

[DrawTextBiDiModeFlagsReadingOn](#)

[EndDrag](#)

[GetControlsAlignment](#)

[GetParentComponent](#)

[GetTextBuf](#)

[GetTextLen](#)

[HasParent](#)

[Hide](#)

[InitiateAction](#)

[IsRightToLeft](#)

[ManualDock](#)

[ManualFloat](#)

[MouseWheelHandler](#)

[ParentToClient](#)

[Perform](#)

[Refresh](#)

[ReplaceDockedControl](#)

[ScreenToClient](#)

[SendToBack](#)

[SetTextBuf](#)

[Show](#)

[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

Derived from TComponent

[BeforeDestruction](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TAdvStringGrid Methods

[TAdvStringGrid](#)

[By object](#)

[Legend](#)

[AddBitButton](#)
[AddBitmap](#)
[AddButton](#)
[AddCheckBox](#)
[AddColumn](#)
[AddComboString](#)
[AddComboStringObject](#)
[AddComment](#)
[AddDataImage](#)
[AddFilePicture](#)
[AddIcon](#)
[AddImageIdx](#)
[AddMarker](#)
[AddMultiImage](#)
[AddNode](#)
[AddPicture](#)
[AddProgress](#)
[AddProgressEx](#)
[AddRadio](#)
[AddRotated](#)
[AddRow](#)
[AppendToCSV](#)
[AppendToHTML](#)
[Assign](#)
[AutoNumberCol](#)
[AutoNumberRow](#)
[AutoSizeCells](#)
[AutoSizeCol](#)
[AutoSizeColumns](#)
[AutoSizeRow](#)
[AutoSizeRows](#)
[BaseCell](#)
[BeforeDestruction](#)
[BeginDrag](#)
[BeginUpdate](#)
[BringToFront](#)
[Broadcast](#)
[CalcFooter](#)
[CanFocus](#)
[CellRect](#)
[CellSize](#)
[CellSpan](#)
[CellToReal](#)
[CellToRich](#)
[CheckCell](#)
[CheckCells](#)
[CheckCol](#)
[CheckGrid](#)
[CheckRow](#)
[Clear](#)
[ClearCols](#)
[ClearColumnHeaders](#)
[ClearComboString](#)
[ClearNormalCells](#)
[ClearRect](#)
[ClearRowHeaders](#)
[ClearRows](#)
[ClearRowSelect](#)

[ClearSelectedCells](#)
[ClearSelection](#)
[ClientToParent](#)
[ClientToScreen](#)
[ColSpanIdentical](#)
[ColumnAvg](#)
[ColumnMax](#)
[ColumnMin](#)
[ColumnSum](#)
[ContainsControl](#)
[ContractAll](#)
[ContractNode](#)
[ControlAtPos](#)
[CopySelectionToClipboard](#)
[CopyToClipBoard](#)
[Create](#)
[CreateBitButton](#)
[CreateBitmap](#)
[CreateFilePicture](#)
[CreateIcon](#)
[CreateParented](#)
[CreatePicture](#)
[CreateRadio](#)
[CutSelectionToClipboard](#)
[CutToClipboard](#)
[DefaultHandler](#)
[Destroy](#)
[DestroyComponents](#)
[Destroying](#)
[DisableAlign](#)
[DisplColIndex](#)
[DisplRowIndex](#)
[Dock](#)
[DockDrop](#)
[DoneInplaceEdit](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EnableAlign](#)
[EndDrag](#)
[EndUpdate](#)
[ExecuteAction](#)
[ExpandAll](#)
[ExpandNode](#)
[FilterRow](#)
[Find](#)
[FindChildControl](#)
[FindComponent](#)
[FindFirst](#)
[FindNext](#)
[FlipChildren](#)
[Focused](#)
[FreeNotification](#)
[FreeOnRelease](#)
[FullCell](#)
[GetBitmap](#)
[GetCheckBoxState](#)
[GetComboCount](#)
[GetControlsAlignment](#)
[GetFilePicture](#)
[GetImageIdx](#)

[GetMarker](#)
[GetNamePath](#)
[GetNodeSpan](#)
[GetNodeSpanType](#)
[GetNodeState](#)
[GetParentComponent](#)
[GetParentRow](#)
[GetPicture](#)
[GetRadioIdx](#)
[GetRadioStrings](#)
[GetRealCol](#)
[GetRealRow](#)
[GetTabOrderList](#)
[GetTextBuf](#)
[GetTextLen](#)
[GetVersionNr](#)
[GetVersionString](#)
[Group](#)
[GroupAvg](#)
[GroupCalc](#)
[GroupCount](#)
[GroupMax](#)
[GroupMin](#)
[GroupSum](#)
[HandleAllocated](#)
[HandleNeeded](#)
[HasButton](#)
[HasCheckBox](#)
[HasDataCheckBox](#)
[HasParent](#)
[Hide](#)
[HideCellEdit](#)
[HideColumn](#)
[HideColumns](#)
[HideInplaceEdit](#)
[HideRow](#)
[HideRows](#)
[HideRowsEx](#)
[HideSelectedRows](#)
[HideSelection](#)
[HideUnSelectedRows](#)
[HilightInCell](#)
[HilightInCol](#)
[HilightInGrid](#)
[HilightInRow](#)
[HilightText](#)
[InitiateAction](#)
[InitSortXRef](#)
[InsertChildRow](#)
[InsertCols](#)
[InsertComponent](#)
[InsertControl](#)
[InsertFromCSV](#)
[InsertRows](#)
[Invalidate](#)
[IsBaseCell](#)
[IsCell](#)
[IsComment](#)
[IsHiddenColumn](#)
[IsHiddenRow](#)
[IsImplementorOf](#)
[IsMergedCell](#)

[IsNode](#)
[IsRadio](#)
[IsRightToLeft](#)
[IsRotated](#)
[IsSelected](#)
[IsXMergedCell](#)
[IsYMergedCell](#)
[LoadAtPointFromBinStream](#)
[LoadColSizes](#)
[LoadFromBinFile](#)
[LoadFromBinStream](#)
[LoadFromCSV](#)
[LoadFromFile](#)
[LoadFromFixed](#)
[LoadFromMDBSQL](#)
[LoadFromMDBTable](#)
[LoadFromStream](#)
[LoadFromXLS](#)
[LoadFromXLSSheet](#)
[LoadPrintSettings](#)
[ManualDock](#)
[ManualFloat](#)
[MapFontHeight](#)
[MapFontSize](#)
[MarkInCell](#)
[MarkInCol](#)
[MarkInGrid](#)
[MarkInRow](#)
[MarkText](#)
[MaxRowSpan](#)
[MergeCells](#)
[MergeCols](#)
[MergeColumnCells](#)
[MergeRowCells](#)
[MinRowSpan](#)
[MouseCoord](#)
[MouseToCell](#)
[MouseWheelHandler](#)
[MoveColumn](#)
[MoveRow](#)
[NumHiddenColumns](#)
[NumHiddenRows](#)
[PaintTo](#)
[ParentToClient](#)
[PasteFromClipboard](#)
[PasteSelectionFromClipboard](#)
[Perform](#)
[Print](#)
[PrintDraw](#)
[PrintDrawRect](#)
[PrintPreview](#)
[PrintPreviewRect](#)
[PrintPreviewSelectedRows](#)
[PrintRect](#)
[PrintSelectedRows](#)
[PushButton](#)
[QSort](#)
[QSortGroup](#)
[QSortIndexed](#)
[QuickSort](#)
[QuickSortIndexed](#)
[RandomFill](#)

[RealColIndex](#)
[Realign](#)
[RealRowIndex](#)
[ReferenceInterface](#)
[Refresh](#)
[RegisterNotifier](#)
[RemoveBitmap](#)
[RemoveButton](#)
[RemoveCheckBox](#)
[RemoveChildRow](#)
[RemoveCols](#)
[RemoveComboString](#)
[RemoveComment](#)
[RemoveComponent](#)
[RemoveControl](#)
[RemoveDataImage](#)
[RemoveDuplicates](#)
[RemoveFilePicture](#)
[RemoveFreeNotification](#)
[RemoveIcon](#)
[RemoveImageIdx](#)
[RemoveMarker](#)
[RemoveMultiImage](#)
[RemoveNode](#)
[RemovePicture](#)
[RemoveProgress](#)
[RemoveRadio](#)
[RemoveRotated](#)
[RemoveRows](#)
[RemoveRowsEx](#)
[RemoveSelectedRows](#)
[RemoveUnSelectedRows](#)
[Repaint](#)
[RepaintCell](#)
[RepaintCol](#)
[RepaintRect](#)
[RepaintRow](#)
[ReplaceDockedControl](#)
[ResetUpdate](#)
[Resize](#)
[RichToCell](#)
[RowAvg](#)
[RowMax](#)
[RowMin](#)
[RowSpanIdentical](#)
[RowSum](#)
[SafeCallException](#)
[SaveColSizes](#)
[SavePrintSettings](#)
[SaveRectToBinStream](#)
[SaveToASCII](#)
[SaveToBinFile](#)
[SaveToBinStream](#)
[SaveToCSV](#)
[SaveToDOC](#)
[SaveToFile](#)
[SaveToFixed](#)
[SaveToHTML](#)
[SaveToStream](#)
[SaveToXLS](#)
[SaveToXLSheet](#)
[SaveToXML](#)

[ScaleBy](#)
[ScreenToCell](#)
[ScreenToClient](#)
[ScrollBy](#)
[ScrollInView](#)
[Search](#)
[SelectCols](#)
[SelectedText](#)
[SelectRange](#)
[SelectRows](#)
[SendToBack](#)
[SetBounds](#)
[SetCheckBoxState](#)
[SetComboSelection](#)
[SetComboSelectionString](#)
[SetFocus](#)
[SetNodeSpan](#)
[SetNodeState](#)
[SetRadiIdx](#)
[SetRotated](#)
[SetSubComponent](#)
[SetTextBuf](#)
[Show](#)
[ShowCellEdit](#)
[ShowColumnHeaders](#)
[ShowInplaceEdit](#)
[ShowRowHeaders](#)
[SortByColumn](#)
[SortedRowIndex](#)
[SortLine](#)
[SortSwapRows](#)
[SplitAllCells](#)
[SplitCells](#)
[SplitColumnCells](#)
[SplitRowCells](#)
[StretchRightColumn](#)
[SwapColumns](#)
[SwapRows](#)
[TextFill](#)
[ToggleCheckBox](#)
[UnGroup](#)
[UnHideColumn](#)
[UnHideColumns](#)
[UnHideColumnsAll](#)
[UnHideRow](#)
[UnHideRows](#)
[UnHideRowsAll](#)
[UnHideSelection](#)
[UnHighlightInCell](#)
[UnHighlightInCol](#)
[UnHighlightInGrid](#)
[UnHighlightInRow](#)
[UnHighlightText](#)
[UnMarkInCell](#)
[UnMarkInCol](#)
[UnMarkInGrid](#)
[UnMarkInRow](#)
[UnMarkText](#)
[UnRegisterNotifier](#)
[UnSortedRowIndex](#)
[Update](#)
[UpdateAction](#)

UpdateControlState
UseRightToLeftAlignment
UseRightToLeftReading
UseRightToLeftScrollBar
ValidateCell
Zoom

TAdvStringGrid Properties

[TAdvStringGrid](#) [Alphabetically](#) [Legend](#)

In TAdvStringGrid

- [ActiveCellColor](#)
- [ActiveCellFont](#)
- [ActiveCellShow](#)
- [Alignments](#)
- [AllCells](#)
- [AllColWidths](#)
- [AllObjects](#)
- [AlwaysQuotes](#)
- [AlwaysValidate](#)
- [AnchorHint](#)
- [ArrowColor](#)
- [AutoNumAlign](#)
- [AutoSize](#)
- [BackGround](#)
- [Bands](#)
- [BtnEdit](#)
- [BtnUnitEdit](#)
- [CellChecker](#)
- [CellGraphics](#)
- [CellGraphicSize](#)
- [CellImages](#)
- [CellNode](#)
- [CellTypes](#)
- [CheckFalse](#)
- [CheckTrue](#)
- [ClearTextOnly](#)
- [ColCount](#)
- [Colors](#)
- [ColumnHeaders](#)
- [ColumnSize](#)
- [Combobox](#)
- [CommentColor](#)
- [Compares](#)
- [ControlLook](#)
- [ControlValues](#)
- [CurrentCell](#)
- [Cursor](#)
- [Dates](#)
- [DateTimePicker](#)
- [DefaultRowHeight](#)
- [Delimiter](#)
- [DragDropSettings](#)
- [EditLink](#)
- [EditMode](#)
- [EditWithTags](#)
- [EnableBlink](#)
- [EnableHTML](#)
- [EnableWheel](#)
- [EnhRowColMove](#)
- [EnhTextSize](#)
- [ExcelClipboardFormat](#)
- [ExcelStyleDecimalSeparator](#)
- [FastPrint](#)
- [Filter](#)
- [FilterActive](#)
- [FindBusy](#)
- [FixedAsButtons](#)
- [FixedColAlways](#)

- [FixedCols](#)
- [FixedColWidth](#)
- [FixedFont](#)
- [FixedFooters](#)
- [FixedRightCols](#)
- [FixedRowAlways](#)
- [FixedRowHeight](#)
- [FixedRows](#)
- [Flat](#)
- [FloatFormat](#)
- [FloatingFooter](#)
- [Floats](#)
- [FontColors](#)
- [FontNames](#)
- [FontSizes](#)
- [FontStyles](#)
- [GridImages](#)
- [GroupColumn](#)
- [HideFocusRect](#)
- [HintColor](#)
- [HintShowCells](#)
- [HintShowLargeText](#)
- [HintShowSizing](#)
- [Hovering](#)
- [HTMLHint](#)
- [HTMLSettings](#)
- [ImageCache](#)
- [InplaceRichEdit](#)
- [IntegralHeight](#)
- [IntegralHeigth](#)
- [IntelliPan](#)
- [IntelliZoom](#)
- [Ints](#)
- [JavaCSV](#)
- [LButFlg](#)
- [LockUpdate](#)
- [Look](#)
- [Lookup](#)
- [LookupCaseSensitive](#)
- [LookupHistory](#)
- [LookupItems](#)
- [MaxColWidth](#)
- [MaxEditLength](#)
- [MaxRowHeight](#)
- [MergedColumns](#)
- [MinColWidth](#)
- [MinRowHeight](#)
- [MouseActions](#)
- [Multilinecells](#)
- [Navigation](#)
- [NodeState](#)
- [NormalEdit](#)
- [OemConvert](#)
- [OriginalCellValue](#)
- [PasswordChar](#)
- [PictureContainer](#)
- [PreviewPage](#)
- [PrevSizeX](#)
- [PrevSizeY](#)
- [PrintColEnd](#)
- [PrintColOffset](#)
- [PrintColStart](#)

- [PrintColWidth](#)
- [PrinterDriverFix](#)
- [PrintNrOfPages](#)
- [PrintPageRect](#)
- [PrintPageWidth](#)
- [PrintSettings](#)
- [QuoteEmptyCells](#)
- [RealCol](#)
- [RealRow](#)
- [RichEdit](#)
- [RowCount](#)
- [RowHeaders](#)
- [RowIndicator](#)
- [RowSelect](#)
- [RowSelectCount](#)
- [SaveColCount](#)
- [SaveEndCol](#)
- [SaveEndRow](#)
- [SaveFixedCells](#)
- [SaveHiddenCells](#)
- [SaveRowCount](#)
- [SaveStartCol](#)
- [SaveStartRow](#)
- [ScrollColor](#)
- [ScrollHints](#)
- [ScrollProportional](#)
- [ScrollSynch](#)
- [ScrollType](#)
- [ScrollWidth](#)
- [SelectedCell](#)
- [SelectedCells](#)
- [SelectedCellsCount](#)
- [SelectionColor](#)
- [SelectionRectangle](#)
- [SelectionRectangleColor](#)
- [SelectionResizer](#)
- [SelectionRTFKeep](#)
- [SelectionTextColor](#)
- [ShowNullDates](#)
- [ShowSelection](#)
- [SizeGrowOnly](#)
- [SizeWhileTyping](#)
- [SizeWithForm](#)
- [SortIndexes](#)
- [Sortlist](#)
- [SortSettings](#)
- [SortTime](#)
- [SpinEdit](#)
- [StrippedCells](#)
- [Swaps](#)
- [Times](#)
- [TMSGradientFrom](#)
- [TMSGradientTo](#)
- [UnSortedCells](#)
- [URLColor](#)
- [URLEdit](#)
- [URLFull](#)
- [URLShow](#)
- [UseHTMLHints](#)
- [VAlignment](#)
- [VersionNr](#)
- [VersionString](#)

- [WordWrap](#)
[XYOffset](#)

Derived from TBaseGrid

[Cells](#)

- [DefaultDrawing](#)
[GraphicObjects](#)
- [GridLineColor](#)
- [GridLineWidth](#)
[Objects](#)

Derived from TStringGrid

[Cols](#)

[Rows](#)

Derived from TDrawGrid

- [Align](#)
- [Anchors](#)
- [BiDiMode](#)
- [BorderStyle](#)
- [Color](#)
- [Constraints](#)
- [Ctl3D](#)
- [DefaultColWidth](#)
- [DragCursor](#)
- [DragKind](#)
- [DragMode](#)
- [Enabled](#)
- [FixedColor](#)
- [Font](#)
- [Options](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentCtl3D](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [PopupMenu](#)
- [ScrollBars](#)
- [ShowHint](#)
- [TabOrder](#)
- [Visible](#)
- [VisibleColCount](#)
- [VisibleRowCount](#)

Derived from TCustomDrawGrid

- [Canvas](#)
- [Col](#)
- [ColWidths](#)
- [EditorMode](#)
- [GridHeight](#)
- [GridWidth](#)
- [LeftCol](#)
- [Row](#)
- [RowHeights](#)
- [Selection](#)
- [TabStops](#)
- [TopRow](#)

Derived from TCustomGrid

- [TabStop](#)

Derived from TWinControl

- [AlignDisabled](#)
- [Brush](#)
[class](#)
- [ControlCount](#)
- [Controls](#)
- [DockClientCount](#)
- [DockClients](#)
- [DoubleBuffered](#)
- [Handle](#)
- [ParentWindow](#)
- [Showing](#)
- [VisibleDockClientCount](#)

Derived from TControl

- [Action](#)
- [BoundsRect](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [ControlState](#)
- [ControlStyle](#)
- [DockOrientation](#)
- [Floating](#)
- [FloatingDockSiteClass](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LRDockWidth](#)
- [Parent](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [Width](#)
- [WindowProc](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TAdvStringGrid Properties

[TAdvStringGrid](#)

[By object](#)

[Legend](#)

[Action](#)

- [ActiveCellColor](#)
- [ActiveCellFont](#)
- [ActiveCellShow](#)
- [Align](#)
- [AlignDisabled](#)
- [Alignments](#)
- [AllCells](#)
- [AllColWidths](#)
- [AllObjects](#)

[AlwaysQuotes](#)

[AlwaysValidate](#)

- [AnchorHint](#)
- [Anchors](#)
- [ArrowColor](#)
- [AutoNumAlign](#)
- [AutoSize](#)
- [BackGround](#)
- [Bands](#)
- [BiDiMode](#)
- [BorderStyle](#)
- [BoundsRect](#)
- [Brush](#)
- [BtnEdit](#)
- [BtnUnitEdit](#)
- [Canvas](#)
- [CellChecker](#)
- [CellGraphics](#)
- [CellGraphicSize](#)
- [CellImages](#)
- [CellNode](#)
- [Cells](#)
- [CellTypes](#)
- [CheckFalse](#)
- [CheckTrue](#)
- [class](#)
- [ClearTextOnly](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [Col](#)
- [ColCount](#)
- [Color](#)
- [Colors](#)
- [Cols](#)
- [ColumnHeaders](#)
- [ColumnSize](#)

[ColWidths](#)

- [Combobox](#)
- [CommentColor](#)
- [ComObject](#)
- [Compares](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)

- [Constraints](#)
- [ControlCount](#)
- [ControlLook](#)
- [Controls](#)
- [ControlState](#)
- [ControlStyle](#)
- [ControlValues](#)
- [Ctl3D](#)
- [CurrentCell](#)
- [Cursor](#)
- [Dates](#)
- [DateTimePicker](#)
- [DefaultColWidth](#)
- [DefaultDrawing](#)
- [DefaultRowHeight](#)
- [Delimiter](#)
- [DesignInfo](#)
- [DockClientCount](#)
- [DockClients](#)
- [DockOrientation](#)
- [DoubleBuffered](#)
- [DragCursor](#)
- [DragDropSettings](#)
- [DragKind](#)
- [DragMode](#)
- [EditLink](#)
- [EditMode](#)
- [EditorMode](#)
- [EditWithTags](#)
- [EnableBlink](#)
- [Enabled](#)
- [EnableHTML](#)
- [EnableWheel](#)
- [EnhRowColMove](#)
- [EnhTextSize](#)
- [ExcelClipboardFormat](#)
- [ExcelStyleDecimalSeparator](#)
- [FastPrint](#)
- [Filter](#)
- [FilterActive](#)
- [FindBusy](#)
- [FixedAsButtons](#)
- [FixedColAlways](#)
- [FixedColor](#)
- [FixedCols](#)
- [FixedColWidth](#)
- [FixedFont](#)
- [FixedFooters](#)
- [FixedRightCols](#)
- [FixedRowAlways](#)
- [FixedRowHeight](#)
- [FixedRows](#)
- [Flat](#)
- [FloatFormat](#)
- [Floating](#)
- [FloatingDockSiteClass](#)
- [FloatingFooter](#)
- [Floats](#)
- [Font](#)
- [FontColors](#)
- [FontNames](#)
- [FontSizes](#)

[FontStyles](#)

[GraphicObjects](#)

[GridHeight](#)

- [GridImages](#)
- [GridLineColor](#)
- [GridLineWidth](#)

[GridWidth](#)

[GroupColumn](#)

- [Handle](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [HideFocusRect](#)
- [Hint](#)
- [HintColor](#)
- [HintShowCells](#)
- [HintShowLargeText](#)
- [HintShowSizing](#)
- [HostDockSite](#)
- [Hovering](#)
- [HTMLHint](#)
- [HTMLSettings](#)
- [ImageCache](#)
- [InplaceRichEdit](#)
- [IntegralHeight](#)
- [IntegralHeighth](#)
- [IntelliPan](#)
- [IntelliZoom](#)
- [Ints](#)
- [JavaCSV](#)
- [LButFlg](#)
- [Left](#)
- [LeftCol](#)

[LockUpdate](#)

- [Look](#)
- [Lookup](#)
- [LookupCaseSensitive](#)
- [LookupHistory](#)
- [LookupItems](#)
- [LRDockWidth](#)
- [MaxColWidth](#)
- [MaxEditLength](#)
- [MaxRowHeight](#)
- [MergedColumns](#)

[MinColWidth](#)

[MinRowHeight](#)

- [MouseActions](#)
- [Multilinecells](#)
- [Name](#)
- [Navigation](#)
- [NodeState](#)
- [NormalEdit](#)
- [Objects](#)
- [OemConvert](#)
- [Options](#)
- [OriginalCellValue](#)
- [Owner](#)
- [Parent](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentCtl3D](#)

- [ParentFont](#)
- [ParentShowHint](#)
- [ParentWindow](#)
- [PasswordChar](#)
- [PictureContainer](#)
- [PopupMenu](#)
- [PreviewPage](#)
- [PrevSizeX](#)
- [PrevSizeY](#)
- [PrintColEnd](#)
- [PrintColOffset](#)
- [PrintColStart](#)
- [PrintColWidth](#)
- [PrinterDriverFix](#)
- [PrintNrOfPages](#)
- [PrintPageRect](#)
- [PrintPageWidth](#)
- [PrintSettings](#)
- [QuoteEmptyCells](#)
- [RealCol](#)
- [RealRow](#)
- [RichEdit](#)
- [Row](#)
- [RowCount](#)
- [RowHeaders](#)
- [RowHeights](#)
- [RowIndicator](#)
- [Rows](#)
- [RowSelect](#)
- [RowSelectCount](#)
- [SaveColCount](#)
- [SaveEndCol](#)
- [SaveEndRow](#)
- [SaveFixedCells](#)
- [SaveHiddenCells](#)
- [SaveRowCount](#)
- [SaveStartCol](#)
- [SaveStartRow](#)
- [ScrollBars](#)
- [ScrollColor](#)
- [ScrollHints](#)
- [ScrollProportional](#)
- [ScrollSynch](#)
- [ScrollType](#)
- [ScrollWidth](#)
- [SelectedCell](#)
- [SelectedCells](#)
- [SelectedCellsCount](#)
- [Selection](#)
- [SelectionColor](#)
- [SelectionRectangle](#)
- [SelectionRectangleColor](#)
- [SelectionResizer](#)
- [SelectionRTFKeep](#)
- [SelectionTextColor](#)
- [ShowHint](#)
- [Showing](#)
- [ShowNullDates](#)
- [ShowSelection](#)
- [SizeGrowOnly](#)
- [SizeWhileTyping](#)
- [SizeWithForm](#)

- [SortIndexes](#)
- [Sortlist](#)
- [SortSettings](#)
- [SortTime](#)
- [SpinEdit](#)
- [StrippedCells](#)
- [Swaps](#)
- [TabOrder](#)
- [TabStop](#)
- [TabStops](#)
- [Tag](#)
- [TBDockHeight](#)
- [Times](#)
- [TMSGradientFrom](#)
- [TMSGradientTo](#)
- [Top](#)
- [TopRow](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [UnSortedCells](#)
- [URLColor](#)
- [URLEdit](#)
- [URLFull](#)
- [URLShow](#)
- [UseHTMLHints](#)
- [VAlignment](#)
- [VCLComObject](#)
- [VersionNr](#)
- [VersionString](#)
- [Visible](#)
- [VisibleColCount](#)
- [VisibleDockClientCount](#)
- [VisibleRowCount](#)
- [Width](#)
- [WindowProc](#)
- [WordWrap](#)
- [XYOffset](#)

TAnchorClickEvent

Unit

advgrid

TAnchorClickEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; Anchor: **string**; **var** AutoHandle: Boolean) **of object**;

TAnchorEvent

Unit

advgrid

TAnchorEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; Anchor: **string**) **of object**;

TAnchorHintEvent

Unit

advgrid

TAnchorHintEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; var Anchor: **string**) **of object**;

TAutoAddRowEvent

Unit

advgrid

TAutoAddRowEvent=**procedure**(Sender: TObject; ARow: **Integer**) **of object**;

TAutoDeleteRowEvent

Unit

advgrid

TAutoDeleteRowEvent=**procedure**(Sender: TObject; ARow: **Integer**) **of object**;

TAutoInsertColEvent

Unit

advgrid

TAutoInsertColEvent=**procedure**(Sender: TObject; ACol: **Integer**) **of object**;

TAutoInsertRowEvent

Unit

advgrid

TAutoInsertRowEvent=**procedure**(Sender: TObject; ARow: **Integer**) of object;

TBackGround

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Sets the grid background. This bitmap can be tiled across all cells as background bitmap or positioned.

TBackGround.Bitmap

[TBackGround](#)

property Bitmap: [TBitmap](#); **Description**
Contains the background bitmap

TBackGround.Cells

[TBackGround](#)

property Cells: [TBackgroundCells](#); **Description**

Selects to which cells the background applies.

TBackgroundCells = (bcNormal,bcFixed)

When the background only applies to the fixed cells, the display should be tiled.

TBackGround.Display

[TBackGround](#)

property Display: [TBackgroundDisplay](#); **Description**

Display mode can be tiled (bdTile) or on a fixed position (bdFixed)

TBackGround.Left

[TBackGround](#)

property Left: **Integer**; **Description**

Coordinate from left of the bitmap when fixed position is used.

TBackGround.Top

[TBackGround](#)

property Top: **Integer**; **Description**

Coordinate from top of the bitmap when fixed position is used.

TBackGround Events

[TBackGround](#)

[Alphabetically](#)

[Legend](#)

TBackGround Events

[TBackGround](#)

[.By object](#)

[Legend](#)

TBackGround Methods

[TBackGround](#) [Alphabetically](#)

[Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TBackGround Methods

TBackGround

.By object

Legend

Assign

Destroy

GetNamePath

TBackGround Properties

[TBackGround](#) [Alphabetically](#) [Legend](#)

In TBackGround

- [Bitmap](#)
- [Cells](#)
- [Display](#)
- [Left](#)
- [Top](#)

TBackGround Properties

[TBackGround](#)

[_By object](#)

[Legend](#)

- [Bitmap](#)
- [Cells](#)
- [Display](#)
- [Left](#)
- [Top](#)

TBackGroundCells

Unit

advgrid

TBackGroundCells=(bcNormal, bcFixed, bcAll); **Description**

Value

Meaning

bcNormal Background bitmap is only used for normal cells

bcFixed Background bitmap is only used for all cells including fixed cells

bcAll Background bitmap is used for all cells

TBackGroundDisplay

Unit

advgrid

TBackGroundDisplay=(bdTile, bdFixed); **Description**

Value

Meaning

bdTile Background bitmap is tiled

bdFixed Background bitmap is fixed

TBands

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Controls the appearance of bands in the grid.

TBands.Active

[TBands](#)

property Active: Boolean **default** False; **Description**

When true, the band settings are used to set the background color of rows in the grid.

TBands.PrimaryColor

[TBands](#)

property PrimaryColor: TColor; **Description**
Background color of the primary bands

TBands.PrimaryLength

[TBands](#)

property PrimaryLength: **Integer**; **Description**
Number of successive rows in the primary band.

TBands.Print

[TBands](#)

property Print: Boolean; **Description**

When true, the banding colors are also used for printing.

TBands.SecondaryColor

[TBands](#)

property SecondaryColor: TColor; **Description**
Background color of the secondary bands

TBands.SecondaryLength

[TBands](#)

property SecondaryLength: **Integer**; **Description**
Number of successive rows in the secondary band.

TBands Events

[TBands](#) [Alphabetically](#)

[Legend](#)

TBands Events

[TBands](#) [By object](#)

[Legend](#)

TBands Methods

[TBands](#) [Alphabetically](#) [Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TBands Methods

[TBands](#) [By object](#)

[Legend](#)

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TBands Properties

[TBands](#) [Alphabetically](#) [Legend](#)

In TBands

- [Active](#)
- [PrimaryColor](#)
- [PrimaryLength](#)
- [Print](#)
- [SecondaryColor](#)
- [SecondaryLength](#)

TBands Properties

[TBands](#) [By object](#) [Legend](#)

- [Active](#)
- [PrimaryColor](#)
- [PrimaryLength](#)
- [Print](#)
- [SecondaryColor](#)
- [SecondaryLength](#)

TBaseGrid

[Hierarchy](#)

[Properties](#)

[Methods](#) [Events](#)

Unit

BaseGrid

Description

Base grid class

TBaseGrid.OnGetDisplText

[TBaseGrid](#)

TGetDisplTextEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; var Value: **string**) of object;
property OnGetDisplText: [TGetDisplTextEvent](#);

Description

Queries the text to be displayed for a given cell.

This allows to override to display the default text in Cells[Col,Row] and use a virtual text instead.

TBaseGrid.BaseCell

[TBaseGrid](#)

function BaseCell(ACol, ARow: **Integer**): TPoint;

Description

Returns the cell coordinate of the base cell (top / left cell) of a merged cell

TBaseGrid.CellRect

[TBaseGrid](#)

function CellRect(c, r: **Integer**): TRect;

Description

Returns the rectangle in pixel coordinates of the cell

TBaseGrid.CellSize

[TBaseGrid](#)

function CellSize(c, r: **Integer**): TPoint;

Description

Returns width and height in pixels of a cell.

TBaseGrid.CellSpan

[TBaseGrid](#)

function CellSpan(ACol, ARow: **Integer**): TPoint;

Description

Returns the width and height in cell coordinates of the merged cell.

TBaseGrid.ColSpanIdentical

[TBaseGrid](#)

function ColSpanIdentical(ACol1, ACol2: **Integer**): Boolean;

TBaseGrid.FullCell

[TBaseGrid](#)

function FullCell(c, r: **Integer**): TRect;

Description

Returns the full rectangle in cell coordinates of a merged cell

TBaseGrid.IsBaseCell

[TBaseGrid](#)

function IsBaseCell(ACol, ARow: **Integer**): Boolean;

Description

When true, the cell is the base cell (top / left cell) of a merged cell.

TBaseGrid.IsMergedCell

[TBaseGrid](#)

function IsMergedCell(ACol, ARow: **Integer**): Boolean;

Description

Returns true if the cell is a merged cell

TBaseGrid.IsXMergedCell

[TBaseGrid](#)

function IsXMergedCell(ACol, ARow: **Integer**): Boolean;

Description

Returns true when the cell is merged in the horizontal direction

TBaseGrid.IsYMergedCell

[TBaseGrid](#)

function IsYMergedCell(ACol, ARow: **Integer**): Boolean;

Description

Returns true when the cell is merged in the vertical direction

TBaseGrid.MaxRowSpan

[TBaseGrid](#)

function MaxRowSpan(ARow: **Integer**): **Integer**;

Description

Returns the max. row span of merged cells in the row ARow

TBaseGrid.MergeCells

[TBaseGrid](#)

procedure MergeCells(c, r, spanx, spany: **Integer**);

Description

Merges cells starting at coordinate [c,r] for spanx cells in horizontal direction and spany cells in vertical direction. Minimum spanx / spany value is 1.

TBaseGrid.MinRowSpan

[TBaseGrid](#)

function MinRowSpan(ARow: **Integer**): **Integer**;

Description

Returns the min. row span of merged cells in the row ARow

TBaseGrid.RepaintCell

[TBaseGrid](#)

procedure RepaintCell(c, r: **Integer**);

Description

Repaints the cell

TBaseGrid.RowSpanIdentical

[TBaseGrid](#)

function RowSpanIdentical(ARow1, ARow2: **Integer**): Boolean;

TBaseGrid.SplitCells

[TBaseGrid](#)

procedure SplitCells(c, r: **Integer**);

Description

Removes cell merging for the cell

TBaseGrid.CellProperties

[TBaseGrid](#)

property CellProperties[c, r: **Integer**]: [TCellProperties](#);

TBaseGrid.Cells

[TBaseGrid](#)

property Cells[c, r: **Integer**]: **String**;

Description

Sets the cell contents as string

TBaseGrid.CustomSelect

[TBaseGrid](#)

property CustomSelect: Boolean;

TBaseGrid.DefaultDrawing

[TBaseGrid](#)

property DefaultDrawing: Boolean;

TBaseGrid.GraphicObjects

[TBaseGrid](#)

property GraphicObjects[c, r: **Integer**]: TObject;

TBaseGrid.GridLineColor

[TBaseGrid](#)

property GridLineColor: TColor;

Description

Sets the color of grid lines.

TBaseGrid.GridLineWidth

[TBaseGrid](#)

property GridLineWidth: **Integer**;

Description

Sets the width of gridlines

TBaseGrid.GridObjects

[TBaseGrid](#)

property GridObjects[c, r: **Integer**]: TObject;

TBaseGrid.HideLastRow

[TBaseGrid](#)

property HideLastRow: Boolean;

TBaseGrid.NormalRowCount

[TBaseGrid](#)

property NormalRowCount: **Integer**;

TBaseGrid.Objects

[TBaseGrid](#)

property Objects[c, r: **Integer**]: TObject;

Description

Provides access to the Objects that can be stored with each cell in the grid

TBaseGrid.PaintID

[TBaseGrid](#)

property PaintID: **Integer**;

TBaseGrid Events

[TBaseGrid](#)

[Alphabetically](#)

[Legend](#)

In TBaseGrid

- [OnGetDisplText](#)

Derived from TDrawGrid

- [OnClick](#)
- [OnColumnMoved](#)
- [OnContextPopup](#)
- [OnDbClick](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnDrawCell](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnGetEditMask](#)
- [OnGetEditText](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnMouseWheelDown](#)
- [OnMouseWheelUp](#)
- [OnRowMoved](#)
- [OnSelectCell](#)
- [OnSetEditText](#)
- [OnStartDock](#)
- [OnStartDrag](#)
- [OnTopLeftChanged](#)

Derived from TWinControl

Derived from TControl

TBaseGrid Events

TBaseGrid

By object

Legend

- [OnClick](#)
- [OnColumnMoved](#)
- [OnContextPopup](#)
- [OnDbClick](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnDrawCell](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnGetDisplText](#)
- [OnGetEditMask](#)
- [OnGetEditText](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnMouseWheelDown](#)
- [OnMouseWheelUp](#)
- [OnRowMoved](#)
- [OnSelectCell](#)
- [OnSetEditText](#)
- [OnStartDock](#)
- [OnStartDrag](#)
- [OnTopLeftChanged](#)

TBaseGrid Methods

[TBaseGrid](#)

[Alphabetically](#)

[Legend](#)

In TBaseGrid

[BaseCell](#)
[CellRect](#)
[CellSize](#)
[CellSpan](#)
[ColSpanIdentical](#)
[FullCell](#)
[IsBaseCell](#)
[IsMergedCell](#)
[IsXMergedCell](#)
[IsYMergedCell](#)
[MaxRowSpan](#)
[MergeCells](#)
[MinRowSpan](#)
[RepaintCell](#)
[RowSpanIdentical](#)
[SplitCells](#)

Derived from TStringGrid

[Create](#)
[Destroy](#)

Derived from TCustomDrawGrid

[MouseToCell](#)

Derived from TCustomGrid

[MouseCoord](#)

Derived from TCustomControl

Derived from TWinControl

[Broadcast](#)
[CanFocus](#)
[ContainsControl](#)
[ControlAtPos](#)
[CreateParented](#)
[DefaultHandler](#)
[DisableAlign](#)
[DockDrop](#)
[EnableAlign](#)
[FindChildControl](#)
[FlipChildren](#)
[Focused](#)
[GetTabOrderList](#)
[HandleAllocated](#)
[HandleNeeded](#)
[InsertControl](#)
[Invalidate](#)
[PaintTo](#)
[Realign](#)
[RemoveControl](#)
[Repaint](#)
[ScaleBy](#)
[ScrollBy](#)
[SetBounds](#)
[SetFocus](#)
[Update](#)
[UpdateControlState](#)

Derived from TControl

[BeginDrag](#)
[BringToFront](#)
[ClientToParent](#)
[ClientToScreen](#)
[Dock](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EndDrag](#)
[GetControlsAlignment](#)
[GetParentComponent](#)
[GetTextBuf](#)
[GetTextLen](#)
[HasParent](#)
[Hide](#)
[InitiateAction](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[ParentToClient](#)
[Perform](#)
[Refresh](#)
[ReplaceDockedControl](#)
[ScreenToClient](#)
[SendToBack](#)
[SetTextBuf](#)
[Show](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

Derived from TComponent

[BeforeDestruction](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TBaseGrid Methods

TBaseGrid

By object

Legend

Assign
BaseCell
BeforeDestruction
BeginDrag
BringToFront
Broadcast
CanFocus
CellRect
CellSize
CellSpan
ClientToParent
ClientToScreen
ColSpanIdentical
ContainsControl
ControlAtPos
Create
CreateParented
DefaultHandler
Destroy
DestroyComponents
Destroying
DisableAlign
Dock
DockDrop
DragDrop
Dragging
DrawTextBiDiModeFlags
DrawTextBiDiModeFlagsReadingOn
EnableAlign
EndDrag
ExecuteAction
FindChildControl
FindComponent
FlipChildren
Focused
FreeNotification
FreeOnRelease
FullCell
GetControlsAlignment
GetNamePath
GetParentComponent
GetTabOrderList
GetTextBuf
GetTextLen
HandleAllocated
HandleNeeded
HasParent
Hide
InitiateAction
InsertComponent
InsertControl
Invalidate
IsBaseCell
IsImplementorOf
IsMergedCell
IsRightToLeft
IsXMergedCell
IsYMergedCell

[ManualDock](#)
[ManualFloat](#)
[MaxRowSpan](#)
[MergeCells](#)
[MinRowSpan](#)
[MouseCoord](#)
[MouseToCell](#)
[MouseWheelHandler](#)
[PaintTo](#)
[ParentToClient](#)
[Perform](#)
[Realign](#)
[ReferenceInterface](#)
[Refresh](#)
[RemoveComponent](#)
[RemoveControl](#)
[RemoveFreeNotification](#)
[Repaint](#)
[RepaintCell](#)
[ReplaceDockedControl](#)
[RowSpanIdentical](#)
[SafeCallException](#)
[ScaleBy](#)
[ScreenToClient](#)
[ScrollBy](#)
[SendToBack](#)
[SetBounds](#)
[SetFocus](#)
[SetSubComponent](#)
[SetTextBuf](#)
[Show](#)
[SplitCells](#)
[Update](#)
[UpdateAction](#)
[UpdateControlState](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

TBaseGrid Properties

[TBaseGrid](#)

[Alphabetically](#)

[Legend](#)

In TBaseGrid

- [CellProperties](#)
- [Cells](#)
- [CustomSelect](#)
- [DefaultDrawing](#)
- [GraphicObjects](#)
- [GridLineColor](#)
- [GridLineWidth](#)
- [GridObjects](#)
- [HideLastRow](#)
- [NormalRowCount](#)
- [Objects](#)
- [PaintID](#)

Derived from TStringGrid

- [Cols](#)
- [Rows](#)

Derived from TDrawGrid

- [Align](#)
- [Anchors](#)
- [BiDiMode](#)
- [BorderStyle](#)
- [ColCount](#)
- [Color](#)
- [Constraints](#)
- [Ctl3D](#)
- [DefaultColWidth](#)
- [DefaultRowHeight](#)
- [DragCursor](#)
- [DragKind](#)
- [DragMode](#)
- [Enabled](#)
- [FixedColor](#)
- [FixedCols](#)
- [FixedRows](#)
- [Font](#)
- [Options](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentCtl3D](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [PopupMenu](#)
- [RowCount](#)
- [ScrollBars](#)
- [ShowHint](#)
- [TabOrder](#)
- [Visible](#)
- [VisibleColCount](#)
- [VisibleRowCount](#)

Derived from TCustomDrawGrid

- [Canvas](#)
- [Col](#)
- [ColWidths](#)
- [EditorMode](#)
- [GridHeight](#)

[GridWidth](#)
[LeftCol](#)
[Row](#)
[RowHeights](#)
[Selection](#)
[TabStops](#)
[TopRow](#)

Derived from TCustomGrid

- [TabStop](#)

Derived from TWinControl

- [AlignDisabled](#)
- [Brush](#)
[class](#)
- [ControlCount](#)
- [Controls](#)
- [DockClientCount](#)
- [DockClients](#)
- [DoubleBuffered](#)
- [Handle](#)
[ParentWindow](#)
- [Showing](#)
- [VisibleDockClientCount](#)

Derived from TControl

- [Action](#)
- [BoundsRect](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
[ClientWidth](#)
[ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
[DockOrientation](#)
- [Floating](#)
[FloatingDockSiteClass](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
[HostDockSite](#)
- [Left](#)
[LRDockWidth](#)
[Parent](#)
- [TBDockHeight](#)
- [Top](#)
[UndockHeight](#)
[UndockWidth](#)
- [Width](#)
[WindowProc](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
[ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)

- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TBaseGrid Properties

TBaseGrid

By object

Legend

Action

- Align
- AlignDisabled
- Anchors
- BiDiMode
- BorderStyle
- BoundsRect
- Brush
- Canvas
- CellProperties
- Cells
- class
- ClientHeight
- ClientOrigin
- ClientRect

ClientWidth

Col

- ColCount
- Color
- Cols
- ColWidths
- ComObject
- ComponentCount

ComponentIndex

- Components
- ComponentState
- ComponentStyle
- Constraints
- ControlCount
- Controls
- ControlState
- ControlStyle
- Ctl3D
- Cursor
- CustomSelect
- DefaultColWidth
- DefaultDrawing
- DefaultRowHeight
- DesignInfo
- DockClientCount
- DockClients
- DockOrientation
- DoubleBuffered
- DragCursor
- DragKind
- DragMode
- EditorMode
- Enabled
- FixedColor
- FixedCols
- FixedRows
- Floating
- FloatingDockSiteClass
- Font
- GraphicObjects
- GridHeight
- GridLineColor
- GridLineWidth

- [GridObjects](#)
- [GridWidth](#)
- [Handle](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [HideLastRow](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LeftCol](#)
- [LRDockWidth](#)
- [Name](#)
- [NormalRowCount](#)
- [Objects](#)
- [Options](#)
- [Owner](#)
- [PaintID](#)
- [Parent](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentCtl3D](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [ParentWindow](#)
- [PopupMenu](#)
- [Row](#)
- [RowCount](#)
- [RowHeights](#)
- [Rows](#)
- [ScrollBars](#)
- [Selection](#)
- [ShowHint](#)
- [Showing](#)
- [TabOrder](#)
- [TabStop](#)
- [TabStops](#)
- [Tag](#)
- [TBDockHeight](#)
- [Top](#)
- [TopRow](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [VCLComObject](#)
- [Visible](#)
- [VisibleColCount](#)
- [VisibleDockClientCount](#)
- [VisibleRowCount](#)
- [Width](#)
- [WindowProc](#)

TBeforeCellPasteEvent

Unit

advgrid

TBeforeCellPasteEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; var Value: **string**; var Allow: Boolean) **of object**;

TBoolArray

Unit

advgrid

TBoolArray=**array**[0..MAXCOLUMNS]of Boolean;

TButtonClickEvent

Unit

advgrid

TButtonClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**) of object;

TCanAddRowEvent

Unit

advgrid

TCanAddRowEvent=**procedure**(Sender: TObject; **var** CanAdd: Boolean) **of object**;

TCanDeleteRowEvent

Unit

advgrid

TCanDeleteRowEvent=**procedure**(Sender: TObject; ARow: **Integer**; **var** CanDelete: Boolean) **of object**;

TCanEditCellEvent

Unit

advgrid

TCanEditCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; **var** CanEdit: Boolean) **of object**;

TCanInsertRowEvent

Unit

advgrid

TCanInsertRowEvent=**procedure**(Sender: TObject; ARow: **Integer**; **var** CanInsert: Boolean) **of object**;

TCanSortEvent

Unit

advgrid

TCanSortEvent=**procedure**(Sender: TObject; ACol: **Integer**; var DoSort: Boolean) **of object**;

TCellAlignment

[Properties](#)

Unit

AdvGrid

TCellAlignment.Alignment

TCellAlignment

Alignment: TAlignment;

TCellAlignment.VAlignment

TCellAlignment

VAlignment: TVAAlignment;

TCellAlignment Properties

[TCellAlignment](#) [Alphabetically](#) [Legend](#)

In TCellAlignment

[Alignment](#)
[VAlignment](#)

TCellAlignment Properties

[TCellAlignment](#)

[By object](#)

[Legend](#)

[Alignment](#)

[VAlignment](#)

TCellBorder

Unit

advgrid

TCellBorder=(cbTop, cbLeft, cbRight, cbBottom); **Description**

Value

Meaning

cbTop	Top side of cell has a border
cbLeft	Left side of cell has a border
cbRight	Right side of cell has a border
cbBottom	Bottom side of cell has a border

TCellBorders

Unit

advgrid

TCellBorders=**set of** [TCellBorder](#); **Description**
Specifies which of the four borders to draw

TCellChangingEvent

Unit

advgrid

TCellChangingEvent=**procedure**(Sender: TObject; OldRow, OldCol, NewRow, NewCol: **Integer**; var Allow: Boolean) **of object**;

TCellComboControlEvent

Unit

advgrid

TCellComboControlEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; CtrlID, CtrlType, CtrlVal: **string**; Values: TStringList; **var** Edit: Boolean; **var** DropCount: **Integer**) of object;

TCellControlEvent

Unit

advgrid

TCellControlEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; CtrlID, CtrlType, CtrlVal: **string**) of **object**;

TCellGraphic

[Hierarchy](#) [Properties](#)

[Methods](#)

Unit

advgrid

TCellGraphic.Assign

[TCellGraphic](#)

procedure Assign(Source: [TPersistent](#));

TCellGraphic.GetPictureSize

[TCellGraphic](#)

function GetPictureSize(cw, rh: **Integer**; hastext: Boolean): TPoint;

TCellGraphic.SetAngle

[TCellGraphic](#)

```
procedure SetAngle(AAngle: smallint);
```

TCellGraphic.SetBitButton

[TCellGraphic](#)

procedure SetBitButton(bw, bh: **Integer**; caption: **string**; Glyph: [TBitmap](#); hal: [TCellHAlign](#); val: [TCellVAlign](#));

TCellGraphic.SetBitmap

[TCellGraphic](#)

procedure SetBitmap(ABmp: [TBitmap](#); Transparent: Boolean; hal: [TCellHAlign](#); val: [TCellVAlign](#));

TCellGraphic.SetButton

[TCellGraphic](#)

procedure SetButton(bw, bh: **Integer**; caption: **string**; hal: [TCellHAlign](#); val: [TCellVAlign](#));

TCellGraphic.SetCheckBox

[TCellGraphic](#)

procedure SetCheckBox(Value, Data, Flat: Boolean; hal: [TCellHAlign](#); val: [TCellVAlign](#));

TCellGraphic.SetDataImage

[TCellGraphic](#)

procedure SetDataImage(idx: **Integer**; hal: [TCellHAlign](#); val: [TCellVAlign](#));

TCellGraphic.SetFilePicture

[TCellGraphic](#)

procedure SetFilePicture(APicture: TFilePicture; Transparent: Boolean; stretchmode: [TStretchMode](#); padding: **Integer**; hal: [TCellHAlign](#); val: [TCellVAlign](#));

TCellGraphic.SetIcon

[TCellGraphic](#)

procedure SetIcon(aicon: [ticon](#); hal: [TCellHAlign](#); val: [TCellVAlign](#));

TCellGraphic.SetImageldx

[TCellGraphic](#)

procedure SetImageldx(idx: **Integer**; hal: [TCellHAlign](#); val: [TCellVAlign](#));

TCellGraphic.SetMultiImage

[TCellGraphic](#)

procedure SetMultiImage(Col, Row, dir: **Integer**; hal: [TCellHAlign](#); val: [TCellVAlign](#); Notifier: TImageChangeEvent);

TCellGraphic.SetPicture

[TCellGraphic](#)

procedure SetPicture(APicture: [TPicture](#); Transparent: Boolean; StretchMode: [TStretchMode](#); padding: **Integer**; hal: [TCellHAlign](#); val: [TCellVAlign](#));

TCellGraphic.CellAngle

[TCellGraphic](#)

property CellAngle: Integer;

TCellGraphic.CellBitmap

[TCellGraphic](#)

property CellBitmap: [TBitmap](#);

TCellGraphic.CellBoolean

[TCellGraphic](#)

property CellBoolean: Boolean;

TCellGraphic.CellColor

TCellGraphic

CellColor: TColor;

TCellGraphic.CellCreated

TCellGraphic

property CellCreated: Boolean;

TCellGraphic.CellData

[TCellGraphic](#)

CellData: **Integer**;

TCellGraphic.CellErrFrom

TCellGraphic

property CellErrFrom: **SmallInt**;

TCellGraphic.CellErrLen

[TCellGraphic](#)

property CellErrLen: **SmallInt**;

TCellGraphic.CellHAlign

TCellGraphic

property CellHAlign: TCellHAlign;

TCellGraphic.CellIcon

TCellGraphic

property CellIcon: TIcon;

TCellGraphic.CellIndex

[TCellGraphic](#)

property CellIndex: **Integer**;

TCellGraphic.CellObject

TCellGraphic

CellObject: TObjct;

TCellGraphic.CellText

[TCellGraphic](#)

property CellText: **string**;

TCellGraphic.CellTransparent

[TCellGraphic](#)

property CellTransparent: Boolean;

TCellGraphic.CellType

TCellGraphic

property CellType: TCellType;

TCellGraphic.CellVAlign

TCellGraphic

property CellVAlign: TCellVAlign;

TCellGraphic.CellValue

TCellGraphic

property CellValue: Double;

TCellGraphic.CellVar

[TCellGraphic](#)

property CellVar: Variant;

TCellGraphic Events

[TCellGraphic](#)

[Alphabetically](#)

[Legend](#)

TCellGraphic Events

TCellGraphic

.By object

Legend

TCellGraphic Methods

[TCellGraphic](#)

[Alphabetically](#)

[Legend](#)

In TCellGraphic

[Assign](#)

[GetPictureSize](#)

[SetAngle](#)

[SetBitButton](#)

[SetBitmap](#)

[SetButton](#)

[SetCheckBox](#)

[SetDataImage](#)

[SetFilePicture](#)

[SetIcon](#)

[SetImageIdx](#)

[SetMultiImage](#)

[SetPicture](#)

Derived from TPersistent

[Destroy](#)

[GetNamePath](#)

TCellGraphic Methods

TCellGraphic

By object

Legend

Assign

Destroy

GetNamePath

GetPictureSize

SetAngle

SetBitButton

SetBitmap

SetButton

SetCheckBox

SetDataImage

SetFilePicture

SetIcon

SetImageIdx

SetMultiImage

SetPicture

TCellGraphic Properties

[TCellGraphic](#)

[Alphabetically](#)

[Legend](#)

In TCellGraphic

- [CellAngle](#)
- [CellBitmap](#)
- [CellBoolean](#)
- [CellColor](#)
- [CellCreated](#)
- [CellData](#)
- [CellErrFrom](#)
- [CellErrLen](#)
- [CellHAlign](#)
- [CellIcon](#)
- [CellIndex](#)
- [CellObject](#)
- [CellText](#)
- [CellTransparent](#)
- [CellType](#)
- [CellVAlign](#)
- [CellValue](#)
- [CellVar](#)

TCellGraphic Properties

TCellGraphic

By object

Legend

- CellAngle
- CellBitmap
- CellBoolean
- CellColor
- CellCreated
- CellData
- CellErrFrom
- CellErrLen
- CellHAlign
- CellIcon
- CellIndex
- CellObject
- CellText
- CellTransparent
- CellType
- CellVAlign
- CellValue
- CellVar

TCellHAlign

Unit

advgrid

TCellHAlign=(haLeft, haRight, haCenter, haBeforeText, haAfterText, haFull); **Description**

Value	Meaning
haLeft	Left alignment independent of cell text
haRight	Right alignment independent of cell text
haCenter	Center alignment independent of cell text
haBeforeText	Position image left before text
haAfterText	Position image right after text
haFull	Use full cell width for image

TCellNode

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Controls the appearance of nodes in the grid

TCellNode.Color

[TCellNode](#)

property Color: TColor; **Description**

Sets the background color of the node when display mode is cnFlat

TCellNode.ContractGlyph

[TCellNode](#)

property ContractGlyph: [TBitmap](#); **Description**

Holds the glyph to display when node is in contracted state when NodeType is cnGlyph

TCellNode.ExpandGlyph

[TCellNode](#)

property ExpandGlyph: [TBitmap](#); **Description**

Holds the glyph to display when node is in expanded state when NodeType is cnGlyph

TCellNode.NodeColor

[TCellNode](#)

property NodeColor: TColor; **Description**

Sets the node color when display mode is cnFlat

TCellNode.NodeType

[TCellNode](#)

property NodeType: [TNodeType](#) **default** cnFlat; **Description**
Controls 4 node display modes : cnFlat, cn3D, cnGlyph, cnLeaf

TCellNode.ShowTree

[TCellNode](#)

property ShowTree: Boolean; **Description**

When true, a tree is displayed for the rows within the node span.

TCellNode Events

[TCellNode](#)

[Alphabetically](#)

[Legend](#)

TCellNode Events

TCellNode

.By object

Legend

TCellNode Methods

[TCellNode](#)

[Alphabetically](#)

[Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TCellNode Methods

TCellNode

.By object

Legend

Assign

Destroy

GetNamePath

TCellNode Properties

[TCellNode](#)

[Alphabetically](#)

[Legend](#)

In TCellNode

- [Color](#)
- [ContractGlyph](#)
- [ExpandGlyph](#)
- [NodeColor](#)
- [NodeType](#)
- [ShowTree](#)

TCellNode Properties

TCellNode

.By object

Legend

- Color
- ContractGlyph
- ExpandGlyph
- NodeColor
- NodeType
- ShowTree

TCellProperties

[Hierarchy](#) [Properties](#)

[Methods](#)

Unit

BaseGrid

TCellProperties.Alignment

[TCellProperties](#)

property Alignment: [TAlignment](#);

TCellProperties.BaseCell

[TCellProperties](#)

property BaseCell[c, r: **Integer**]: TPoint;

TCellProperties.BorderColor

[TCellProperties](#)

property BorderColor: TColor;

TCellProperties.BorderWidth

[TCellProperties](#)

property BorderWidth: **integer**;

TCellProperties.BrushColor

[TCellProperties](#)

property BrushColor: TColor;

TCellProperties.BrushStyle

TCellProperties

property BrushStyle: TBrushStyle;

TCellProperties.CellObject

[TCellProperties](#)

property CellObject: TObject;

TCellProperties.CellSpanX

[TCellProperties](#)

property CellSpanX: **Integer**;

TCellProperties.CellSpanY

[TCellProperties](#)

property CellSpanY: **Integer**;

TCellProperties.FontColor

[TCellProperties](#)

property FontColor: TColor;

TCellProperties.FontSize

[TCellProperties](#)

property FontSize: integer;

TCellProperties.FontStyle

TCellProperties

property FontStyle: TFontStyles;

TCellProperties.GraphicObject

[TCellProperties](#)

property GraphicObject: TObject;

TCellProperties.IsBaseCell

[TCellProperties](#)

property IsBaseCell: Boolean;

TCellProperties.PaintID

[TCellProperties](#)

property PaintID: Integer;

TCellProperties.VAlignment

[TCellProperties](#)

property VAlignment: [TVAlignment](#);

TCellProperties.WordWrap

[TCellProperties](#)

property WordWrap: Boolean;

TCellProperties Events

[TCellProperties](#)

[Alphabetically](#)

[Legend](#)

TCellProperties Events

[TCellProperties](#)

[By object](#)

[Legend](#)

TCellProperties Methods

[TCellProperties](#)

[Alphabetically](#)

[Legend](#)

TCellProperties Methods

[TCellProperties](#)

[.By object](#)

[Legend](#)

TCellProperties Properties

[TCellProperties](#) [Alphabetically](#) [Legend](#)

In TCellProperties

- [Alignment](#)
-
- [BaseCell](#)
- [BorderColor](#)
- [BorderWidth](#)
- [BrushColor](#)
- [BrushStyle](#)
- [CellObject](#)
- [CellSpanX](#)
- [CellSpanY](#)
- [FontColor](#)
- [FontSize](#)
- [FontStyle](#)
- [GraphicObject](#)
- [IsBaseCell](#)
- [PaintID](#)
- [VAlignment](#)
- [WordWrap](#)

TCellProperties Properties

[TCellProperties](#)

[By object](#)

[Legend](#)

- [Alignment](#)
- [BaseCell](#)
- [BorderColor](#)
- [BorderWidth](#)
- [BrushColor](#)
- [BrushStyle](#)
- [CellObject](#)
- [CellSpanX](#)
- [CellSpanY](#)
- [FontColor](#)
- [FontSize](#)
- [FontStyle](#)
- [GraphicObject](#)
- [IsBaseCell](#)
- [PaintID](#)
- [VAlignment](#)
- [WordWrap](#)

TCellsChangedEvent

Unit

advgrid

TCellsChangedEvent=**procedure**(Sender: TObject; R: TRect) **of object**;

TCellType

Unit

advgrid

TCellType=(ctBitmap, ctIcon, ctNone, ctImageList, ctCheckBox, ctDataCheckBox, ctRotated, ctDataImage, ctNode, ctRadio, ctEmpty, ctImages, ctPicture, ctFilePicture, ctValue, ctProgress, ctComment, ctButton, ctBitButton); **Description**

Value	Meaning
ctBitmap	Bitmap
ctIcon	Icon
ctNone	No graphic in cell
ctImageList	Imagelist image
ctCheckBox	Checkbox
ctDataCheckBox	Cell text dependent checkbox
ctRotated	Rotated text
ctDataImage	Cell text dependent image
ctNode	Node
ctRadio	Radio button
ctEmpty	No graphic in cell
ctImages	Multiple Imagelist images
ctPicture	Picture
ctFilePicture	File picture
ctValue	Value
ctProgress	Progress bar
ctComment	Comment
ctButton	Button
ctBitButton	BitButton

TCellValidateEvent

Unit

advgrid

TCellValidateEvent=**procedure**(Sender: TObject; Col, Row: **Integer**; var Value: **String**; var Valid: Boolean) **of object**;

TCellVAlign

Unit

advgrid

TCellVAlign=(vaTop, vaBottom, vaCenter, vaUnderText, vaAboveText, vaFull); **Description**

Value

Meaning

vaTop	Top alignment independent of cell text
vaBottom	Bottom alignment independent of cell text
vaCenter	Center alignment independent of cell text
vaUnderText	Position image under text
vaAboveText	Position image above text
vaFull	Use full cell height for image

TCheckBoxClickEvent

Unit

advgrid

TCheckBoxClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; State: Boolean) **of object**;

TClickCellEvent

Unit

advgrid

TClickCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**) of object;

TClickSortEvent

Unit

advgrid

TClickSortEvent=**procedure**(Sender: TObject; ACol: **Integer**) of object;

TClipboardEvent

Unit

advgrid

TClipboardEvent=**procedure**(Sender: TObject; **var** Allow: Boolean) **of object**;

TClipOperation

Unit

advgrid

TClipOperation=(coCut, coCopy); **Description**

Value

Meaning

coCut

The last clipboard operation was a Cut

coCopy

The last clipboard operation was a Copy

TColChangingEvent

Unit

advgrid

TColChangingEvent=**procedure**(Sender: TObject; OldCol, NewCol: **Integer**; var Allow: Boolean) **of**
object;

TColumnSize

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Controls saving and restoring of column widths as well as behaviour of column resizing.

TColumnSize.Key

[TColumnSize](#) [See also Example](#)

property Key: **string**; **Description**

Key in INI file or registry where sizes are stored.

TColumnSize.Key example

Saving the column sizes to the INI file colsizes.ini in a section [sizes] :

```
AdvStringGrid.ColumnSize.Key := '\colsizes.ini';  
AdvStringGrid.ColumnSize.Section = 'sizes';  
AdvStringGrid.SaveColSizes;
```


TColumnSize.Location

TColumnSize

property Location: TColumnSizeLocation; **Description**

Determines whether the column sizes are saved in the registry or in an INI file.

TColumnSize.Save

[TColumnSize](#)

property Save: Boolean **default** False; **Description**

When true, column sizes are saved into registry or INI file.

TColumnSize.Section

[TColumnSize](#) [See also](#)

property Section: **string**; **Description**

Section in INI file or registry where sizes are stored

TColumnSize.Stretch

[TColumnSize](#) [See also](#)

property Stretch: Boolean **default** False; **Description**

Automatic stretching of a column to full grid size. The column to be stretched is set by [StretchColumn](#)

TColumnSize.StretchColumn

[TColumnSize](#) [See also](#)

property StretchColumn: **Integer default** - 1; **Description**

Sets the column that is automatically sized to fill the full grid. If StretchColumn is -1, the last column is automatically sized.

TColumnSize.SynchWithGrid

[TColumnSize](#)

property SynchWithGrid: Boolean **default** False; **Description**

When true, column sizes increase / decrease automatically when the grid resizes

TColumnSize Events

[TColumnSize](#)

[Alphabetically](#)

[Legend](#)

TColumnSize Events

[TColumnSize](#)

[.By object](#)

[Legend](#)

TColumnSize Methods

[TColumnSize](#) [Alphabetically](#)

[Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TColumnSize Methods

[TColumnSize](#)

[.By object](#)

[Legend](#)

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TColumnSize Properties

[TColumnSize](#)

[Alphabetically](#)

[Legend](#)

In TColumnSize

- [Key](#)
- [Location](#)
- [Save](#)
- [Section](#)
- [Stretch](#)
- [StretchColumn](#)
- [SynchWithGrid](#)

TColumnSize Properties

TColumnSize

.By object

Legend

- Key
- Location
- Save
- Section
- Stretch
- StretchColumn
- SynchWithGrid

TColumnSizeEvent

Unit

advgrid

TColumnSizeEvent=**procedure**(Sender: TObject; ACol: **Integer**; **var** Allow: Boolean) **of object**;

TColumnSizeLocation

Unit

advgrid

TColumnSizeLocation=(clRegistry, clIniFile); **Description**

Value

Meaning

clRegistry	Uses the registry to store column widths
clIniFile	Uses an INI file to store the column widths

TComboChangeEvent

Unit

advgrid

TComboChangeEvent=**procedure**(Sender: TObject; ACol, ARow, AItemIndex: **Integer**; ASelection: **string**) of object;

TComboObjectChangeEvent

Unit

advgrid

TComboObjectChangeEvent=**procedure**(Sender: TObject; ACol, ARow, AItemIndex: **Integer**; ASelection: **string**; AObject: TObject) **of object**;

TControlCombo

Hierarchy _____

Unit

advgrid

TControlCombo Events

[TControlCombo](#)

[Alphabetically](#)

[Legend](#)

TControlCombo Events

[TControlCombo](#)

[.By object](#)

[Legend](#)

TControlCombo Methods

[TControlCombo](#)

[Alphabetically](#)

[Legend](#)

TControlCombo Methods

[TControlCombo](#)

[.By object](#)

[Legend](#)

TControlCombo Properties

[TControlCombo](#)

[Alphabetically](#)

[Legend](#)

TControlCombo Properties

[TControlCombo](#)

[.By object](#)

[Legend](#)

TControlEdit

[Hierarchy](#)

[Properties](#)

[Methods](#) [Events](#)

Unit

advgrid

TControlEdit Events

[TControlEdit](#) [Alphabetically](#)

[Legend](#)

Derived from TEdit

- [OnChange](#)
- [OnClick](#)
- [OnContextPopup](#)
- [OnDbClick](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnStartDock](#)
- [OnStartDrag](#)

Derived from TWinControl

Derived from TControl

TControlEdit Events

TControlEdit

By object

Legend

- [OnChange](#)
- [OnClick](#)
- [OnContextPopup](#)
- [OnDbClick](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnStartDock](#)
- [OnStartDrag](#)

TControlEdit Methods

[TControlEdit](#) [Alphabetically](#) [Legend](#)

Derived from TCustomEdit

[Clear](#)
[ClearSelection](#)
[ClearUndo](#)
[CopyToClipboard](#)
[Create](#)
[CutToClipboard](#)
[DefaultHandler](#)
[GetSelTextBuf](#)
[PasteFromClipboard](#)
[SelectAll](#)
[SetSelTextBuf](#)
[Undo](#)

Derived from TWinControl

[Broadcast](#)
[CanFocus](#)
[ContainsControl](#)
[ControlAtPos](#)
[CreateParented](#)
[Destroy](#)
[DisableAlign](#)
[DockDrop](#)
[EnableAlign](#)
[FindChildControl](#)
[FlipChildren](#)
[Focused](#)
[GetTabOrderList](#)
[HandleAllocated](#)
[HandleNeeded](#)
[InsertControl](#)
[Invalidate](#)
[PaintTo](#)
[Realign](#)
[RemoveControl](#)
[Repaint](#)
[ScaleBy](#)
[ScrollBy](#)
[SetBounds](#)
[SetFocus](#)
[Update](#)
[UpdateControlState](#)

Derived from TControl

[BeginDrag](#)
[BringToFront](#)
[ClientToParent](#)
[ClientToScreen](#)
[Dock](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EndDrag](#)
[GetControlsAlignment](#)
[GetParentComponent](#)
[GetTextBuf](#)
[GetTextLen](#)
[HasParent](#)

[Hide](#)
[InitiateAction](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[ParentToClient](#)
[Perform](#)
[Refresh](#)
[ReplaceDockedControl](#)
[ScreenToClient](#)
[SendToBack](#)
[SetTextBuf](#)
[Show](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

Derived from TComponent

[BeforeDestruction](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TControlEdit Methods

TControlEdit

By object

Legend

Assign
BeforeDestruction
BeginDrag
BringToFront
Broadcast
CanFocus
Clear
ClearSelection
ClearUndo
ClientToParent
ClientToScreen
ContainsControl
ControlAtPos
CopyToClipboard
Create
CreateParented
CutToClipboard
DefaultHandler
Destroy
DestroyComponents
Destroying
DisableAlign
Dock
DockDrop
DragDrop
Dragging
DrawTextBiDiModeFlags
DrawTextBiDiModeFlagsReadingOn
EnableAlign
EndDrag
ExecuteAction
FindChildControl
FindComponent
FlipChildren
Focused
FreeNotification
FreeOnRelease
GetControlsAlignment
GetNamePath
GetParentComponent
GetSelTextBuf
GetTabOrderList
GetTextBuf
GetTextLen
HandleAllocated
HandleNeeded
HasParent
Hide
InitiateAction
InsertComponent
InsertControl
Invalidate
IsImplementorOf
IsRightToLeft
ManualDock
ManualFloat
MouseWheelHandler
PaintTo

[ParentToClient](#)
[PasteFromClipboard](#)
[Perform](#)
[Realign](#)
[ReferenceInterface](#)
[Refresh](#)
[RemoveComponent](#)
[RemoveControl](#)
[RemoveFreeNotification](#)
[Repaint](#)
[ReplaceDockedControl](#)
[SafeCallException](#)
[ScaleBy](#)
[ScreenToClient](#)
[ScrollBy](#)
[SelectAll](#)
[SendToBack](#)
[SetBounds](#)
[SetFocus](#)
[SetSelTextBuf](#)
[SetSubComponent](#)
[SetTextBuf](#)
[Show](#)
[Undo](#)
[Update](#)
[UpdateAction](#)
[UpdateControlState](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

TControlEdit Properties

[TControlEdit](#)

[Alphabetically](#)

[Legend](#)

Derived from TEdit

- [Anchors](#)
- [AutoSelect](#)
- [AutoSize](#)
- [BevelEdges](#)
- [BevelInner](#)
- [BevelKind](#)
- [BevelOuter](#)
- [BiDiMode](#)
- [BorderStyle](#)
- [CharCase](#)
- [Color](#)
- [Constraints](#)
- [Ctl3D](#)
- [DragCursor](#)
- [DragKind](#)
- [DragMode](#)
- [Enabled](#)
- [Font](#)
- [HideSelection](#)
- [ImeMode](#)
- [ImeName](#)
- [MaxLength](#)
- [OEMConvert](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentCtl3D](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [PasswordChar](#)
- [PopupMenu](#)
- [ReadOnly](#)
- [ShowHint](#)
- [TabOrder](#)
- [TabStop](#)
- [Text](#)
- [Visible](#)

Derived from TCustomEdit

- [CanUndo](#)
- [Modified](#)
- [SelLength](#)
- [SelStart](#)
- [SelText](#)

Derived from TWinControl

- [AlignDisabled](#)
- [Brush](#)
- [class](#)
- [ControlCount](#)
- [Controls](#)
- [DockClientCount](#)
- [DockClients](#)
- [DoubleBuffered](#)
- [Handle](#)
- [ParentWindow](#)
- [Showing](#)
- [VisibleDockClientCount](#)

Derived from TControl

- [Action](#)
- [Align](#)
- [BoundsRect](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
- [DockOrientation](#)
- [Floating](#)
- [FloatingDockSiteClass](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LRDockWidth](#)
- [Parent](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [Width](#)
- [WindowProc](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TControlEdit Properties

TControlEdit

By object

Legend

Action

Align

- AlignDisabled
- Anchors
- AutoSelect
- AutoSize
- BevelEdges
- BevelInner
- BevelKind
- BevelOuter
- BiDiMode
- BorderStyle
- BoundsRect
- Brush
- CanUndo
- CharCase
- class
- ClientHeight
- ClientOrigin
- ClientRect
- ClientWidth
- Color
- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- Constraints
- ControlCount
- Controls
- ControlState
- ControlStyle
- Ctl3D
- Cursor
- DesignInfo
- DockClientCount
- DockClients
- DockOrientation
- DoubleBuffered
- DragCursor
- DragKind
- DragMode
- Enabled
- Floating
- FloatingDockSiteClass
- Font
- Handle
- Height
- HelpContext
- HelpKeyword
- HelpType
- HideSelection
- Hint
- HostDockSite
- ImeMode
- ImeName
- Left

- [LRDockWidth](#)
- [MaxLength](#)
- [Modified](#)
- [Name](#)
- [OEMConvert](#)
- [Owner](#)
- [Parent](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentCtl3D](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [ParentWindow](#)
- [PasswordChar](#)
- [PopupMenu](#)
- [ReadOnly](#)
- [SelLength](#)
- [SelStart](#)
- [SelText](#)
- [ShowHint](#)
- [Showing](#)
- [TabOrder](#)
- [TabStop](#)
- [Tag](#)
- [TBDockHeight](#)
- [Text](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [VCLComObject](#)
- [Visible](#)
- [VisibleDockClientCount](#)
- [Width](#)
- [WindowProc](#)

TControlLook

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Controls the look of various inplace edit controls

TControlLook.CheckedGlyph

[TControlLook](#)

property CheckedGlyph: [TBitmap](#); **Description**

Sets the glyph for a checked checkbox with the ControlStyle = csGlyph

TControlLook.CheckSize

[TControlLook](#)

property CheckSize: **Integer**; **Description**

Sets the size of a checkbox

TControlLook.Color

[TControlLook](#)

property Color: TColor; **Description**

Sets the color of checkbox or radiobutton in Borland style

TControlLook.ControlStyle

[TControlLook](#)

property ControlStyle: [TControlStyle](#); **Description**

Sets the style of the inplace edit controls checkbox and radiobutton

TControlLook.FlatButton

[TControlLook](#)

property FlatButton: Boolean; **Description**

When true, inplace button controls are displayed in flat mode

TControlLook.RadioOffGlyph

[TControlLook](#)

property RadioOffGlyph: [TBitmap](#); **Description**

Sets the glyph for an unchecked radiobutton with the ControlStyle = csGlyph

TControlLook.RadioOnGlyph

[TControlLook](#)

property RadioOnGlyph: [TBitmap](#); **Description**

Sets the glyph for a checked radiobutton with the ControlStyle = csGlyph

TControlLook.RadioSize

[TControlLook](#)

property RadioSize: **Integer**; **Description**

Sets the size of a radiobutton

TControlLook.UnCheckedGlyph

[TControlLook](#)

property UnCheckedGlyph: [TBitmap](#); **Description**

Sets the glyph for an unchecked checkbox with the ControlStyle = csGlyph

TControlLook Events

[TControlLook](#)

[Alphabetically](#)

[Legend](#)

TControlLook Events

[TControlLook](#)

[.By object](#)

[Legend](#)

TControlLook Methods

[TControlLook](#) [Alphabetically](#)

[Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TControlLook Methods

TControlLook

.By object

Legend

Assign

Destroy

GetNamePath

TControlLook Properties

[TControlLook](#)

[Alphabetically](#)

[Legend](#)

In TControlLook

- [CheckedGlyph](#)
- [CheckSize](#)
- [Color](#)
- [ControlStyle](#)
- [FlatButton](#)
- [RadioOffGlyph](#)
- [RadioOnGlyph](#)
- [RadioSize](#)
- [UnCheckedGlyph](#)

TControlLook Properties

TControlLook

By object

Legend

- CheckedGlyph
- CheckSize
- Color
- ControlStyle
- FlatButton
- RadioOffGlyph
- RadioOnGlyph
- RadioSize
- UnCheckedGlyph

TControlStyle

Unit

advgrid

TControlStyle=(csClassic, csFlat, csWinXP, csBorland, csTMS, csGlyph, csTheme); **Description**

Value	Meaning
csClassic	Classic Windows control look
csFlat	Flat Windows control look
csWinXP	Emulated Windows XP look
csBorland	Borland look
csTMS	TMS look
csGlyph	Glyph
csTheme	Real Windows XP theme look

TCustomCellDrawEvent

Unit

advgrid

TCustomCellDrawEvent=**procedure**(Sender: TObject; Canvas: [TCanvas](#); ACol, ARow: **Integer**; AState: [TGridDrawState](#); ARect: TRect; Printing: Boolean) **of object**;

TCustomCellSizeEvent

Unit

advgrid

TCustomCellSizeEvent=**procedure**(Sender: TObject; Canvas: [TCanvas](#); ACol, ARow: **Integer**; var ASize: TPoint; Printing: Boolean) **of object**;

TCustomCompareEvent

Unit

advgrid

TCustomCompareEvent=**procedure**(Sender: TObject; str1, str2: **string**; var Res: **Integer**) of object;

TDateTimeSpinClickEvent

Unit

advgrid

TDateTimeSpinClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; AValue: TDateTime;
UpDown: Boolean) **of object**;

TDbIClickCellEvent

Unit

advgrid

TDbIClickCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**) **of object**;

TDoFitToPageEvent

Unit

advgrid

TDoFitToPageEvent=**procedure**(Sender: TObject; **var** ScaleFactor: Double; **var** Allow: Boolean) **of**
object;

TDragDropSettings

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Controls various behaviour settings for OLE drag & drop operations

TDragDropSettings.OleAcceptFiles

[TDragDropSettings](#)

property OleAcceptFiles: Boolean **default** False; **Description**
When true, the control accepts OLE file drag & drop operations

TDragDropSettings.OleAcceptText

[TDragDropSettings](#)

property OleAcceptText: Boolean **default** False; **Description**
When true, the control accepts OLE text drag & drop operations

TDragDropSettings.OleColumnDragDrop

[TDragDropSettings](#)

property OleColumnDragDrop: Boolean **default** False;

TDragDropSettings.OleCopyAlways

[TDragDropSettings](#)

property OleCopyAlways: Boolean **default** False;

TDragDropSettings.OleDropRTF

[TDragDropSettings](#)

property OleDropRTF: Boolean **default** False; **Description**
When true, enables full rich text drag and drop.

TDragDropSettings.OleDropSource

[TDragDropSettings](#)

property OleDropSource: Boolean **default** False; **Description**

When true, the grid allows that a range of cells is dragged into another application that supports OLE drag & drop (like Excel / Word) The range of cells needs to be selected first in the grid and then dragged by clicking near the border of the selected cells (just like in Excel).

TDragDropSettings.OleDropTarget

[TDragDropSettings](#)

property OleDropTarget: Boolean **default** False; **Description**

When true, the grid accepts data being dropped on the grid (from Excel, Word) or a CSV file from Explorer. Select a range of cells in Excel or another OLE drag & drop source, and drag into the grid. The data is inserted from the position where the drop is performed. If a filename from Explorer is dropped into the grid, the grid will attempt to open the file if it's a CSV file.

TDragDropSettings.OleEntireRows

[TDragDropSettings](#)

property OleEntireRows: Boolean **default** False; **Description**

When true, entire rows are copied in OLE drag & drop operations (including fixed rows)

TDragDropSettings.OleInsertRows

[TDragDropSettings](#)

property OleInsertRows: Boolean **default** False; **Description**

When true, rows are automatically inserted during drag & drop operations.

TDragDropSettings.OleRemoveRows

[TDragDropSettings](#)

property OleRemoveRows: Boolean **default** False; **Description**

When true, rows are automatically removed during OLE drag & drop move operations

TDragDropSettings Events

[TDragDropSettings](#) [Alphabetically](#)

[Legend](#)

TDragDropSettings Events

[TDragDropSettings](#) [By object](#)

[Legend](#)

TDragDropSettings Methods

[TDragDropSettings](#) [Alphabetically](#)

[Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TDragDropSettings Methods

[TDragDropSettings](#) [By object](#)

[Legend](#)

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TDragDropSettings Properties

[TDragDropSettings](#) [Alphabetically](#)

[Legend](#)

In TDragDropSettings

- [OleAcceptFiles](#)
- [OleAcceptText](#)
- [OleColumnDragDrop](#)
- [OleCopyAlways](#)
- [OleDropRTE](#)
- [OleDropSource](#)
- [OleDropTarget](#)
- [OleEntireRows](#)
- [OleInsertRows](#)
- [OleRemoveRows](#)

TDragDropSettings Properties

[TDragDropSettings](#) [By object](#)

[Legend](#)

- [OleAcceptFiles](#)
- [OleAcceptText](#)
- [OleColumnDragDrop](#)
- [OleCopyAlways](#)
- [OleDropRTE](#)
- [OleDropSource](#)
- [OleDropTarget](#)
- [OleEntireRows](#)
- [OleInsertRows](#)
- [OleRemoveRows](#)

TEditLink

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Base class for an edit link with which any TWinControl descendant editor can be used as inplace editor for the grid.

TEditLink.CreateEditor

[TEditLink](#)

procedure CreateEditor(AParent: [TWinControl](#)); **Description**
Override this method to create the inplace editor

TEditLink.DestroyEditor

[TEditLink](#)

procedure DestroyEditor; **Description**

Override this method to destroy the inplace editor

TEditLink.EditExit

[TEditLink](#)

procedure EditExit(Sender: TObject); **Description**

Override this method to handle leaving focus from the inplace editor

TEditLink.EditKeyDown

[TEditLink](#)

procedure EditKeyDown(Sender: TObject; **var** Key: **Word**; Shift: [TShiftState](#)); **Description**
Override this method to handle the keydown event of the inplace editor

TEditLink.FormExit

[TEditLink](#)

procedure FormExit(Sender: TObject); **Description**

Override this method to handle leaving focus from a popup edit control.

TEditLink.GetCellValue

[TEditLink](#)

function GetCellValue: **string**; **Description**

Retrieves the value of the edited cell

TEditLink.GetEditControl

[TEditLink](#)

function GetEditControl: [TWinControl](#); **Description**

Retrieves the current inplace editor as a TWinControl. Cast to the correct class type to handle special properties of the descendant class.

TEditLink.GetEditorValue

[TEditLink](#)

function GetEditorValue: **string**; **Description**

Retrieves the value of the inplace edit control.

TEditLink.GetParent

[TEditLink](#)

function GetParent: [TWinControl](#); **Description**

Retrieves the parent of the inplace edit control. This is either the grid or the popup window control used for popup style inplace editor.

TEditLink.HideEditor

[TEditLink](#)

procedure HideEditor; **Description**

Hides the inplace editor.

TEditLink.SetCellProps

[TEditLink](#)

procedure SetCellProps(AColor: TColor; AFont: [TFont](#)); **Description**

Override this method to set editor properties according to cell properties such as background color and font before editing starts

TEditLink.SetCellValue

[TEditLink](#)

procedure SetCellValue(s: **string**); **Description**

Sets the value of the cell being edited.

TEditLink.SetEditorValue

[TEditLink](#)

procedure SetEditorValue(s: **string**); **Description**

Sets the value of the current inplace editor

TEditLink.SetFocus

[TEditLink](#)

procedure SetFocus(Value: Boolean); **Description**

Override this method to handle special settings when the inplace editor gains focus.

TEditLink.SetProperties

[TEditLink](#)

procedure SetProperties; **Description**

Override this method to set properties before the inplace editor is displayed.

TEditLink.SetRect

[TEditLink](#)

procedure SetRect(r: TRect); **Description**

Override this method to set the bounding rectangle of the inplace editor

TEditLink.SetVisible

[TEditLink](#)

procedure SetVisible(Value: Boolean); **Description**

Override this method to handle special settings when the inplace editor is set visible

TEditLink.AutoPopupWidth

[TEditLink](#)

property AutoPopupWidth: Boolean; **Description**

When true, the width of the popup is automatically equal to the width of the cell edited.

TEditLink.EditCell

[TEditLink](#)

property EditCell: TPoint; **Description**

Holds the coordinate of the current cell being edited

TEditLink.EditStyle

[TEditLink](#)

property EditStyle: [TEditStyle](#); **Description**

Specifies either an inplace editor that is positioned inside the cell or a popup editor.

TEditLink.Grid

[TEditLink](#)

property Grid: [TAdvStringGrid](#); **Description**

Parent grid of the inplace edit control

TEditLink.PopupHeight

[TEditLink](#)

property PopupHeight: **Integer**; **Description**

Sets the height of a popup edit control (used when EditStyle = esPopup)

TEditLink.PopupWidth

[TEditLink](#)

property PopupWidth: **Integer**; **Description**

Sets the width of a popup edit control (used when EditStyle = esPopup)

TEditLink.Tag

[TEditLink](#)

property Tag: **Integer**; **Description**

Tag has no predefined meaning. The Tag property is provided for the convenience of developers. It can be used for storing an additional integer value or it can be typecast to any 32-bit value such as a component reference or a pointer.

TEditLink.WantKeyEscape

[TEditLink](#)

property WantKeyEscape: Boolean; **Description**

When true, pressing the escape key is handled by the inplace editor. When false, the escape key is handled by the grid.

TEditLink.WantKeyHomeEnd

[TEditLink](#)

property WantKeyHomeEnd: Boolean; **Description**

When true, pressing the home/end keys are handled by the inplace editor. When false, the home/end keys are handled by the grid.

TEditLink.WantKeyLeftRight

[TEditLink](#)

property WantKeyLeftRight: Boolean; **Description**

When true, pressing the left/right keys are handled by the inplace editor. When false, the left/right keys are handled by the grid.

TEditLink.WantKeyPriorNext

[TEditLink](#)

property WantKeyPriorNext: Boolean; **Description**

When true, pressing the prior/next keys are handled by the inplace editor. When false, the prior/next keys are handled by the grid.

TEditLink.WantKeyReturn

[TEditLink](#)

property WantKeyReturn: Boolean; **Description**

When true, pressing the return keys is handled by the inplace editor. When false, the return key is handled by the grid.

TEditLink.WantKeyUpDown

[TEditLink](#)

property WantKeyUpDown: Boolean; **Description**

When true, pressing the up/down keys are handled by the inplace editor. When false, the up/down keys are handled by the grid.

TEditLink Events

[TEditLink](#) [Alphabetically](#)

[Legend](#)

TEditLink Events

[TEditLink](#) [By object](#)

[Legend](#)

TEditLink Methods

[TEditLink](#) [Alphabetically](#)

[Legend](#)

In TEditLink

[CreateEditor](#)
[DestroyEditor](#)
[EditExit](#)
[EditKeyDown](#)
[FormExit](#)
[GetCellValue](#)
[GetEditControl](#)
[GetEditorValue](#)
[GetParent](#)
[HideEditor](#)
[SetCellProps](#)
[SetCellValue](#)
[SetEditorValue](#)
[SetFocus](#)
[SetProperties](#)
[SetRect](#)
[SetVisible](#)

Derived from TComponent

[BeforeDestruction](#)
[Create](#)
[Destroy](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[GetParentComponent](#)
[HasParent](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TEditLink Methods

[TEditLink](#) [By object](#)

[Legend](#)

[Assign](#)
[BeforeDestruction](#)
[Create](#)
[CreateEditor](#)
[Destroy](#)
[DestroyComponents](#)
[DestroyEditor](#)
[Destroying](#)
[EditExit](#)
[EditKeyDown](#)
[ExecuteAction](#)
[FindComponent](#)
[FormExit](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetCellValue](#)
[GetEditControl](#)
[GetEditorValue](#)
[GetNamePath](#)
[GetParent](#)
[GetParentComponent](#)
[HasParent](#)
[HideEditor](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetCellProps](#)
[SetCellValue](#)
[SetEditorValue](#)
[SetFocus](#)
[SetProperties](#)
[SetRect](#)
[SetSubComponent](#)
[SetVisible](#)
[UpdateAction](#)

TEditLink Properties

[TEditLink](#) [Alphabetically](#)

[Legend](#)

In TEditLink

- [AutoPopupWidth](#)
- [EditCell](#)
- [EditStyle](#)
- [Grid](#)
- [PopupHeight](#)
- [PopupWidth](#)
- [Tag](#)
- [WantKeyEscape](#)
- [WantKeyHomeEnd](#)
- [WantKeyLeftRight](#)
- [WantKeyPriorNext](#)
- [WantKeyReturn](#)
- [WantKeyUpDown](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [VCLComObject](#)

TEditLink Properties

TEditLink By object

Legend

- AutoPopupWidth
- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- DesignInfo
- EditCell
- EditStyle
- Grid
- Name
- Owner
- PopupHeight
- PopupWidth
- Tag
- VCLComObject
- WantKeyEscape
- WantKeyHomeEnd
- WantKeyLeftRight
- WantKeyPriorNext
- WantKeyReturn
- WantKeyUpDown

TEditorType

Unit

AdvGrid

TEditorType=(edNormal, edSpinEdit, edComboEdit, edComboList, edEditBtn, edCheckBox, edDateEdit, edDateEditUpDown, edTimeEdit, edButton, edDataCheckBox, edNumeric, edPositiveNumeric, edFloat, edCapital, edMixedCase, edPassword, edUnitEditBtn, edLowerCase, edUpperCase, edFloatSpinEdit, edTimeSpinEdit, edDateSpinEdit, edNumericEditBtn, edFloatEditBtn, edCustom, edRichEdit);

Description

Value	Meaning
edNormal	Edit cell using edit control
edSpinEdit	Edit cell using spin edit control
edFloatSpinEdit	Edit cell using float spin edit control
edTimeSpinEdit	Edit cell using time spin edit control
edDateSpinEdit	Edit cell using date spin edit control
edComboEdit	Edit cell using default ComboBox control
edComboList	Edit cell using ComboList control
edEditBtn	Edit cell using edit box with button control
edUnitEditBtn	Edit cell using edit box + unit field control
edDateEdit	Edit cell using date validated edit control
edButton	Edit cell using button control
edNumeric	Edit cell using numeric validated edit control
edPositiveNumeric	Edit cell using positive numeric validated edit control
edFloat	Edit cell using floating point validated edit control
edCapital	Edit cell using edit control UPPER CASE only
edMixedCase	Edit cell using text validated edit control
edLowerCase	Edit cell in all lower case characters
edUpperCase	Edit cell in all upper case characters
edTimeEdit	Edit cell as time
edDateEditUpDown	Edit date with updown button
edPassword	Edit as password
edRichEdit	Rich text inplace editor
edCustom	A custom inplace is used. This is set through the <code>EditorType</code> property.

TEditStyle

Unit

advgrid

TEditStyle=(esInplace, esPopup); **Description**

Value

Meaning

esInplace the editor is positioned inside the cell

esPopup the editor is a popup editor

TEllipsClickEvent

Unit

advgrid

TEllipsClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; var S: **string**) of **object**;

TEndColumnSizeEvent

Unit

advgrid

TEndColumnSizeEvent=**procedure**(Sender: TObject; ACol: **Integer**) **of object**;

TEndRowSizeEvent

Unit

advgrid

TEndRowSizeEvent=**procedure**(Sender: TObject; ARow: **Integer**) **of object**;

TFilter

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Collection holding the filter specifiers

TFilter.Add

TFilter

function Add: TFilterData;

TFilter.Insert

TFilter

function Insert(index: Integer): TFilterData;

TFilter.ColumnFilter

[TFilter](#)

property ColumnFilter[Col: **Integer**]: [TFilterData](#);

TFilter.Items

[TFilter](#)

property Items[Index: Integer]: [TFilterData](#); **Description**

Holds the list of filter specifications

TFilter Events

[TFilter](#)

[Alphabetically](#)

[Legend](#)

TFilter Events

[TFilter](#) [By object](#)

[Legend](#)

TFilter Methods

[TFilter](#) [Alphabetically](#)

[Legend](#)

In TFilter

[Add](#)

[Insert](#)

Derived from TCollection

[Assign](#)

[BeginUpdate](#)

[Clear](#)

[Create](#)

[Delete](#)

[Destroy](#)

[EndUpdate](#)

[FindItemID](#)

[GetNamePath](#)

[Owner](#)

Derived from TPersistent

TFilter Methods

[TFilter](#) [By object](#)

[Legend](#)

[Add](#)

[Assign](#)

[BeginUpdate](#)

[Clear](#)

[Create](#)

[Delete](#)

[Destroy](#)

[EndUpdate](#)

[FindItemID](#)

[GetNamePath](#)

[Insert](#)

[Owner](#)

TFilter Properties

[TFilter](#) [Alphabetically](#) [Legend](#)

In TFilter

- [ColumnFilter](#)
[Items](#)

Derived from TCollection

- [Count](#)
- [ItemClass](#)

TFilter Properties

TFilter .By object Legend

- ColumnFilter
- Count
- ItemClass
- Items

TFilterData

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Filter specifier for one column

TFilterData.CaseSensitive

[TFilterData](#)

property CaseSensitive: Boolean **default** True; **Description**
When true, the filter condition is treated case sensitive.

TFilterData.Column

[TFilterData](#)

property Column: **smallint**; **Description**

Sets the column for which the filter applies

TFilterData.Condition

[TFilterData](#)

property Condition: **string**; **Description**

The Condition property can contain a condition in following formats :

substring filtering :

S*, S? : * or ? denote multi or single character matches

>, <, >=, =< : larger than, less than, larger or equal, less or equal (when data is numeric, comparison take place based on numeric data, when a valid date is detected, date comparison is used, otherwise it defaults to alphabetic comparisons)

=, ! : equal or not equal

& : logical AND

^ : logical OR

; : logical OR

Thus, an expression : >20 & <30 is a valid filtering expression as well as !A*

TFilterData Events

[TFilterData](#)

[Alphabetically](#)

[Legend](#)

TFilterData Events

TFilterData

.By object

Legend

TFilterData Methods

[TFilterData](#)

[Alphabetically](#)

[Legend](#)

Derived from TCollectionItem

[Create](#)

[Destroy](#)

[GetNamePath](#)

Derived from TPersistent

[Assign](#)

TFilterData Methods

[TFilterData](#)

[.By object](#)

[Legend](#)

[Assign](#)

[Create](#)

[Destroy](#)

[GetNamePath](#)

TFilterData Properties

[TFilterData](#)

[Alphabetically](#)

[Legend](#)

In TFilterData

- [CaseSensitive](#)
- [Column](#)
- [Condition](#)

Derived from TCollectionItem

- [Collection](#)
- [DisplayName](#)
- [ID](#)
- [Index](#)

TFilterData Properties

TFilterData

.By object

Legend

- CaseSensitive
- Collection
- Column
- Condition
- DisplayName
- ID
- Index

TFindParameters

Unit

advgrid

TFindParameters=(fnMatchCase, fnMatchFull, fnMatchRegular, fnDirectionLeftRight, fnMatchStart, fnFindInCurrentRow, fnFindInCurrentCol, fnIncludeFixed, fnAutoGoto, fnIgnoreHTMLTags, fnBackward, fnIncludeHiddenColumns); **Description**

Value	Meaning
fnMatchCase	match case
fnMatchFull	match full word only
fnMatchRegular	match the expression
fnDirectionLeftRight	search from left to right instead of from top to bottom first
fnMatchStart	match from first letter of cell content only
fnFindInCurrentRow	search in current row only
fnFindInCurrentCol	search in current column only
fnIncludeFixed	search fixed rows and columns too
fnAutoGoto	go to cell when found
fnIgnoreHTMLTags	ignore HTML tags in search
fnBackward	search in inverse direction, ie right to left or bottom to top

TFindParams

Unit

advgrid

TFindParams=**set of** [TFindParameters](#); **Description**

Set of parameters for the FindFirst, FindNext or Find methods

TFitToPage

Unit

advgrid

TFitToPage=(fpNever, fpGrow, fpShrink, fpAlways, fpCustom); **Description**

Value

Meaning

fpNever	No resizing is applied to fit the printout on the page
fpGrow	Print sizes are increased to fit on the page, no shrinking is used
fpShrink	Print sizes are only decreased to fit on the page. No larger printing is used.
fpAlways	Increasing or decreasing is used to fit the grid's printout on the page
fpCustom	The shrink or grow ratio is queried through an event.

TFloatingFooter

[Hierarchy](#)

[Properties](#)

[Methods](#)

[Example](#)

Unit

advgrid

Description

Controls settings of floating footer

TFloatingFooter example

Holds the property settings for the display of a fixed footer in the grid.

TFloatingFooter.Invalidate

[TFloatingFooter](#)

procedure Invalidate;

TFloatingFooter.Color

[TFloatingFooter](#)

property Color: TColor; **Description**

Sets the background color of the floating footer panel

TFloatingFooter.Column

[TFloatingFooter](#)

property Column: **Integer**; **Description**

Sets the column for preview in the footer when fsColumnPreview style is chosen

TFloatingFooter.ColumnCalc

[TFloatingFooter](#) [Example](#)

property ColumnCalc[c: **Integer**]: TColumnCalcType; **Description**
Sets the type of column calculation for a floating footer cell.

TFloatingFooter.ColumnCalc example

// sets the calculation type of the first fixed footer cell to a column sum

```
AdvStringGrid.FloatingFooter.ColumnCalc[1] := acSum;
```

// sets the calculation type of the first fixed footer cell to a column count

```
AdvStringGrid.FloatingFooter.ColumnCalc[2] := acCount;
```

TFloatingFooter.CustomTemplate

[TFloatingFooter](#)

property CustomTemplate: **string**; **Description**

Sets the HTML template for custom preview in floating footer

TFloatingFooter.FooterStyle

[TFloatingFooter](#)

property FooterStyle: [TFooterStyle](#); **Description**

Sets the style of the floating footer

TFloatingFooter.Height

[TFloatingFooter](#) [Example](#)

property Height: **Integer**;

TFloatingFooter.Height example

Sets the height of the fixed footer.

TFloatingFooter.Visible

[TFloatingFooter](#)

property Visible: Boolean; **Description**
When true, the floating footer is visible

TFloatingFooter Events

[TFloatingFooter](#)

[Alphabetically](#)

[Legend](#)

TFloatingFooter Events

[TFloatingFooter](#)

[.By object](#)

[Legend](#)

TFloatingFooter Methods

[TFloatingFooter](#) [Alphabetically](#) [Legend](#)

In TFloatingFooter

[Invalidate](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TFloatingFooter Methods

[TFloatingFooter](#)

[.By object](#)

[Legend](#)

[Assign](#)

[Destroy](#)

[GetNamePath](#)

[Invalidate](#)

TFloatingFooter Properties

[TFloatingFooter](#) [Alphabetically](#) [Legend](#)

In TFloatingFooter

- [Color](#)
- [Column](#)
- [ColumnCalc](#)
- [CustomTemplate](#)
- [FooterStyle](#)
- [Height](#)
- [Visible](#)

TFloatingFooter Properties

[TFloatingFooter](#)

[By object](#)

[Legend](#)

- [Color](#)
- [Column](#)
- [ColumnCalc](#)
- [CustomTemplate](#)
- [FooterStyle](#)
- [Height](#)
- [Visible](#)

TFloatSpinClickEvent

Unit

advgrid

TFloatSpinClickEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; AValue: Double; UpDown: Boolean) **of object**;

TFooterPanel

[Hierarchy](#)

[Properties](#)

[Methods](#) [Events](#)

Unit

advgrid

TFooterPanel Events

[TFooterPanel](#) [Alphabetically](#)

[Legend](#)

Derived from TWinControl

Derived from TControl

TFooterPanel Events

[TFooterPanel](#)

[By object](#)

[Legend](#)

TFooterPanel Methods

[TFooterPanel](#) [Alphabetically](#) [Legend](#)

Derived from TCustomPanel

[Create](#)
[GetControlsAlignment](#)

Derived from TCustomControl

[Destroy](#)

Derived from TWinControl

[Broadcast](#)
[CanFocus](#)
[ContainsControl](#)
[ControlAtPos](#)
[CreateParented](#)
[DefaultHandler](#)
[DisableAlign](#)
[DockDrop](#)
[EnableAlign](#)
[FindChildControl](#)
[FlipChildren](#)
[Focused](#)
[GetTabOrderList](#)
[HandleAllocated](#)
[HandleNeeded](#)
[InsertControl](#)
[Invalidate](#)
[PaintTo](#)
[Realign](#)
[RemoveControl](#)
[Repaint](#)
[ScaleBy](#)
[ScrollBy](#)
[SetBounds](#)
[SetFocus](#)
[Update](#)
[UpdateControlState](#)

Derived from TControl

[BeginDrag](#)
[BringToFront](#)
[ClientToParent](#)
[ClientToScreen](#)
[Dock](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EndDrag](#)
[GetParentComponent](#)
[GetTextBuf](#)
[GetTextLen](#)
[HasParent](#)
[Hide](#)
[InitiateAction](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[ParentToClient](#)
[Perform](#)

[Refresh](#)
[ReplaceDockedControl](#)
[ScreenToClient](#)
[SendToBack](#)
[SetTextBuf](#)
[Show](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

Derived from TComponent

[BeforeDestruction](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TFooterPanel Methods

[TFooterPanel](#) [By object](#) [Legend](#)

[Assign](#)
[BeforeDestruction](#)
[BeginDrag](#)
[BringToFront](#)
[Broadcast](#)
[CanFocus](#)
[ClientToParent](#)
[ClientToScreen](#)
[ContainsControl](#)
[ControlAtPos](#)
[Create](#)
[CreateParented](#)
[DefaultHandler](#)
[Destroy](#)
[DestroyComponents](#)
[Destroying](#)
[DisableAlign](#)
[Dock](#)
[DockDrop](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EnableAlign](#)
[EndDrag](#)
[ExecuteAction](#)
[FindChildControl](#)
[FindComponent](#)
[FlipChildren](#)
[Focused](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetControlsAlignment](#)
[GetNamePath](#)
[GetParentComponent](#)
[GetTabOrderList](#)
[GetTextBuf](#)
[GetTextLen](#)
[HandleAllocated](#)
[HandleNeeded](#)
[HasParent](#)
[Hide](#)
[InitiateAction](#)
[InsertComponent](#)
[InsertControl](#)
[Invalidate](#)
[IsImplementorOf](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[PaintTo](#)
[ParentToClient](#)
[Perform](#)
[Realign](#)
[ReferenceInterface](#)
[Refresh](#)
[RemoveComponent](#)

[RemoveControl](#)
[RemoveFreeNotification](#)
[Repaint](#)
[ReplaceDockedControl](#)
[SafeCallException](#)
[ScaleBy](#)
[ScreenToClient](#)
[ScrollBy](#)
[SendToBack](#)
[SetBounds](#)
[SetFocus](#)
[SetSubComponent](#)
[SetTextBuf](#)
[Show](#)
[Update](#)
[UpdateAction](#)
[UpdateControlState](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

TFooterPanel Properties

[TFooterPanel](#) [Alphabetically](#) [Legend](#)

Derived from TCustomPanel

Derived from TCustomControl

Derived from TWinControl

- [AlignDisabled](#)
- [Brush](#)
[class](#)
- [ControlCount](#)
- [Controls](#)
- [DockClientCount](#)
- [DockClients](#)
- [DoubleBuffered](#)
- [Handle](#)
- [ParentWindow](#)
- [Showing](#)
- [TabOrder](#)
- [TabStop](#)
- [VisibleDockClientCount](#)

Derived from TControl

- [Action](#)
- [Align](#)
- [Anchors](#)
- [BiDiMode](#)
- [BoundsRect](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [Constraints](#)
- [ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
- [DockOrientation](#)
- [Enabled](#)
- [Floating](#)
- [FloatingDockSiteClass](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LRDockWidth](#)
- [Parent](#)
- [ShowHint](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [Visible](#)
- [Width](#)
- [WindowProc](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)

ComponentIndex

- Components
- ComponentState
- ComponentStyle
- DesignInfo
- Name
- Owner
- Tag
- VCLComObject

TFooterPanel Properties

[TFooterPanel](#) [By object](#) [Legend](#)

[Action](#)

[Align](#)

- [AlignDisabled](#)
- [Anchors](#)
- [BiDiMode](#)
- [BoundsRect](#)
- [Brush](#)
- [class](#)

[ClientHeight](#)

- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [Constraints](#)
- [ControlCount](#)
- [Controls](#)
- [ControlState](#)

[ControlStyle](#)

- [Cursor](#)
- [DesignInfo](#)
- [DockClientCount](#)
- [DockClients](#)
- [DockOrientation](#)
- [DoubleBuffered](#)

[Enabled](#)

- [Floating](#)
- [FloatingDockSiteClass](#)
- [Handle](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LRDockWidth](#)
- [Name](#)
- [Owner](#)
- [Parent](#)

[ParentWindow](#)

[ShowHint](#)

- [Showing](#)
- [TabOrder](#)
- [TabStop](#)
- [Tag](#)
- [TBDockHeight](#)
- [Top](#)

[UndockHeight](#)

[UndockWidth](#)

[VCLComObject](#)

[Visible](#)

- [VisibleDockClientCount](#)
- [Width](#)

WindowProc

TFooterStyle

Unit

advgrid

TFooterStyle=(fsFixedLastRow, fsColumnPreview, fsCustomPreview); **Description**

Value

Meaning

fsFixedLastRow	Footer shows last row content
fsColumnPreview	Footer shows column contents of selected row
fsCustomPreview	Footer shows custom data

TGetCheckEvent

Unit

advgrid

TGetCheckEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; var Value: **string**) **of object**;

TGetDisplTextEvent

Unit

BaseGrid

TGetDisplTextEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; **var** Value: **string**) **of object**;

TGetEditorPropEvent

Unit

advgrid

TGetEditorPropEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; AEditLink: [TEditLink](#)) **of object**;

TGetEditorTypeEvent

Unit

advgrid

TGetEditorTypeEvent=**procedure**(Sender: TObject; ACol, ARow: **Integer**; var AEditor: [TEditorType](#)) of **object**;

TGridAlignEvent

Unit

advgrid

TGridAlignEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; **var** HAlign: TAlignment; **var** VAlign: TVAlignment) **of object**;

TGridBMPIO

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridBMPIO.Bitmap

TGridBMPIO

property Bitmap: TBitmap;

TGridBMPIO Events

[TGridBMPIO](#)

[Alphabetically](#)

[Legend](#)

TGridBMPIO Events

[TGridBMPIO](#)

[.By object](#)

[Legend](#)

TGridBMPIO Methods

[TGridBMPIO](#) [Alphabetically](#)

[Legend](#)

Derived from TComponent

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridBMPIO Methods

TGridBMPIO

By object

Legend

Assign

BeforeDestruction

Create

Destroy

DestroyComponents

Destroying

ExecuteAction

FindComponent

FreeNotification

FreeOnRelease

GetNamePath

GetParentComponent

HasParent

InsertComponent

IsImplementorOf

ReferenceInterface

RemoveComponent

RemoveFreeNotification

SafeCallException

SetSubComponent

UpdateAction

TGridBMPIO Properties

[TGridBMPIO](#) [Alphabetically](#)

[Legend](#)

In TGridBMPIO

- [Bitmap](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridBMPIO Properties

[TGridBMPIO](#)

[By object](#)

[Legend](#)

- [Bitmap](#)
- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridBorderEvent

Unit

advgrid

TGridBorderEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; APen: TPen; **var** Borders: TCellBorders) **of object**;

TGridBorderPropEvent

Unit

advgrid

TGridBorderPropEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; LeftPen, TopPen, RightPen, BottomPen: TPen) **of object**;

TGridButton

[Hierarchy](#)

[Properties](#)

[Methods](#) [Events](#)

Unit

advgrid

TGridButton.CreateParams

[TGridButton](#)

```
procedure CreateParams(var Params: TCreateParams);
```

TGridButton Events

[TGridButton](#)

[Alphabetically](#)

[Legend](#)

Derived from TButton

- [OnClick](#)
- [OnContextPopup](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnStartDock](#)
- [OnStartDrag](#)

Derived from TWinControl

Derived from TControl

TGridButton Events

TGridButton

By object

Legend

- OnClick
- OnContextPopup
- OnDragDrop
- OnDragOver
- OnEndDock
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnStartDock
- OnStartDrag

TGridButton Methods

[TGridButton](#) [Alphabetically](#) [Legend](#)

In TGridButton

[CreateParams](#)

Derived from TButton

[Click](#)

[Create](#)

[UseRightToLeftAlignment](#)

Derived from TButtonControl

Derived from TWinControl

[Broadcast](#)

[CanFocus](#)

[ContainsControl](#)

[ControlAtPos](#)

[CreateParented](#)

[DefaultHandler](#)

[Destroy](#)

[DisableAlign](#)

[DockDrop](#)

[EnableAlign](#)

[FindChildControl](#)

[FlipChildren](#)

[Focused](#)

[GetTabOrderList](#)

[HandleAllocated](#)

[HandleNeeded](#)

[InsertControl](#)

[Invalidate](#)

[PaintTo](#)

[Realign](#)

[RemoveControl](#)

[Repaint](#)

[ScaleBy](#)

[ScrollBy](#)

[SetBounds](#)

[SetFocus](#)

[Update](#)

[UpdateControlState](#)

Derived from TControl

[BeginDrag](#)

[BringToFront](#)

[ClientToParent](#)

[ClientToScreen](#)

[Dock](#)

[DragDrop](#)

[Dragging](#)

[DrawTextBiDiModeFlags](#)

[DrawTextBiDiModeFlagsReadingOn](#)

[EndDrag](#)

[GetControlsAlignment](#)

[GetParentComponent](#)

[GetTextBuf](#)

[GetTextLen](#)

[HasParent](#)

[Hide](#)

[InitiateAction](#)

[IsRightToLeft](#)

[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[ParentToClient](#)
[Perform](#)
[Refresh](#)
[ReplaceDockedControl](#)
[ScreenToClient](#)
[SendToBack](#)
[SetTextBuf](#)
[Show](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

Derived from TComponent

[BeforeDestruction](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridButton Methods

TGridButton

By object

Legend

Assign
BeforeDestruction
BeginDrag
BringToFront
Broadcast
CanFocus
Click
ClientToParent
ClientToScreen
ContainsControl
ControlAtPos
Create
CreateParams
CreateParented
DefaultHandler
Destroy
DestroyComponents
Destroying
DisableAlign
Dock
DockDrop
DragDrop
Dragging
DrawTextBiDiModeFlags
DrawTextBiDiModeFlagsReadingOn
EnableAlign
EndDrag
ExecuteAction
FindChildControl
FindComponent
FlipChildren
Focused
FreeNotification
FreeOnRelease
GetControlsAlignment
GetNamePath
GetParentComponent
GetTabOrderList
GetTextBuf
GetTextLen
HandleAllocated
HandleNeeded
HasParent
Hide
InitiateAction
InsertComponent
InsertControl
Invalidate
IsImplementorOf
IsRightToLeft
ManualDock
ManualFloat
MouseWheelHandler
PaintTo
ParentToClient
Perform
Realign
ReferenceInterface

[Refresh](#)
[RemoveComponent](#)
[RemoveControl](#)
[RemoveFreeNotification](#)
[Repaint](#)
[ReplaceDockedControl](#)
[SafeCallException](#)
[ScaleBy](#)
[ScreenToClient](#)
[ScrollBy](#)
[SendToBack](#)
[SetBounds](#)
[SetFocus](#)
[SetSubComponent](#)
[SetTextBuf](#)
[Show](#)
[Update](#)
[UpdateAction](#)
[UpdateControlState](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

TGridButton Properties

[TGridButton](#)

[Alphabetically](#)

[Legend](#)

Derived from TButton

- [Action](#)
- [Anchors](#)
- [BiDiMode](#)
- [Cancel](#)
- [Caption](#)
- [Constraints](#)
- [Default](#)
- [DragCursor](#)
- [DragKind](#)
- [DragMode](#)
- [Enabled](#)
- [Font](#)
- [ModalResult](#)
- [ParentBiDiMode](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [PopupMenu](#)
- [ShowHint](#)
- [TabOrder](#)
- [TabStop](#)
- [Visible](#)

Derived from TButtonControl

Derived from TWinControl

- [AlignDisabled](#)
- [Brush](#)
[class](#)
- [ControlCount](#)
- [Controls](#)
- [DockClientCount](#)
- [DockClients](#)
- [DoubleBuffered](#)
- [Handle](#)
- [ParentWindow](#)
- [Showing](#)
- [VisibleDockClientCount](#)

Derived from TControl

- [Align](#)
- [BoundsRect](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
- [DockOrientation](#)
- [Floating](#)
- [FloatingDockSiteClass](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)

- [LRDockWidth](#)
 - [Parent](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
 - [UndockWidth](#)
- [Width](#)
- [WindowProc](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridButton Properties

TGridButton

By object

Legend

- Action
- Align
- AlignDisabled
- Anchors
- BiDiMode
- BoundsRect
- Brush
- Cancel
- Caption
- class
- ClientHeight
- ClientOrigin
- ClientRect
- ClientWidth
- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- Constraints
- ControlCount
- Controls
- ControlState
- ControlStyle
- Cursor
- Default
- DesignInfo
- DockClientCount
- DockClients
- DockOrientation
- DoubleBuffered
- DragCursor
- DragKind
- DragMode
- Enabled
- Floating
- FloatingDockSiteClass
- Font
- Handle
- Height
- HelpContext
- HelpKeyword
- HelpType
- Hint
- HostDockSite
- Left
- LRDockWidth
- ModalResult
- Name
- Owner
- Parent
- ParentBiDiMode
- ParentFont
- ParentShowHint
- ParentWindow
- PopupMenu
- ShowHint

- [Showing](#)
- [TabOrder](#)
- [TabStop](#)
- [Tag](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [VCLComObject](#)
- [Visible](#)
- [VisibleDockClientCount](#)
- [Width](#)
- [WindowProc](#)

TGridCellIO

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridCellIO.Cell

[TGridCellIO](#)

property Cell: string;

TGridCellIO.Col

TGridCellIO

property Col: **SmallInt;**

TGridCellIO.HasProp

[TGridCellIO](#)

property HasProp: Boolean;

TGridCellIO.Row

[TGridCellIO](#)

property Row: SmallInt;

TGridCellIO Events

[TGridCellIO](#)

[Alphabetically](#)

[Legend](#)

TGridCellIO Events

TGridCellIO

.By object

Legend

TGridCellIO Methods

[TGridCellIO](#) [Alphabetically](#)

[Legend](#)

Derived from TComponent

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridCellIO Methods

TGridCellIO

.By object

Legend

Assign

BeforeDestruction

Create

Destroy

DestroyComponents

Destroying

ExecuteAction

FindComponent

FreeNotification

FreeOnRelease

GetNamePath

GetParentComponent

HasParent

InsertComponent

IsImplementorOf

ReferenceInterface

RemoveComponent

RemoveFreeNotification

SafeCallException

SetSubComponent

UpdateAction

TGridCellIO Properties

[TGridCellIO](#) [Alphabetically](#)

[Legend](#)

In TGridCellIO

- [Cell](#)
- [Col](#)
- [HasProp](#)
- [Row](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridCellIO Properties

TGridCellIO

By object

Legend

- Cell
- Col
- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- DesignInfo
- HasProp
- Name
- Owner
- Row
- Tag
- VCLComObject

TGridCellPropIO

[Hierarchy](#) [Properties](#)

[Methods](#)

Unit

advgrid

TGridCellPropIO.CellProperties

TGridCellPropIO

property CellProperties: TCellProperties;

TGridCellPropIO.HasGraphic

[TGridCellPropIO](#)

property HasGraphic: Boolean;

TGridCellPropIO Events

[TGridCellPropIO](#)

[Alphabetically](#)

[Legend](#)

TGridCellPropIO Events

[TGridCellPropIO](#)

[.By object](#)

[Legend](#)

TGridCellPropIO Methods

[TGridCellPropIO](#) [Alphabetically](#)

[Legend](#)

Derived from TComponent

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridCellPropIO Methods

[TGridCellPropIO](#) [By object](#)

[Legend](#)

[Assign](#)
[BeforeDestruction](#)
[Create](#)
[Destroy](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[GetParentComponent](#)
[HasParent](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

TGridCellPropIO Properties

[TGridCellPropIO](#) [Alphabetically](#) [Legend](#)

In TGridCellPropIO

- [CellProperties](#)
- [HasGraphic](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridCellPropIO Properties

[TGridCellPropIO](#)

[By object](#)

[Legend](#)

- [CellProperties](#)
- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [HasGraphic](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridChangeNotifier

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridChangeNotifier Events

[TGridChangeNotifier](#)

[Alphabetically](#)

[Legend](#)

TGridChangeNotifier Events

TGridChangeNotifier

.By object

Legend

TGridChangeNotifier Methods

[TGridChangeNotifier](#)

[Alphabetically](#)

[Legend](#)

Derived from TComponent

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridChangeNotifier Methods

[TGridChangeNotifier](#)

[By object](#)

[Legend](#)

[Assign](#)

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

TGridChangeNotifier Properties

[TGridChangeNotifier](#)

[Alphabetically](#)

[Legend](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridChangeNotifier Properties

TGridChangeNotifier

By object

Legend

- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- DesignInfo
- Name
- Owner
- Tag
- VCLComObject

TGridCheckBox

[Hierarchy](#) [Properties](#)

[Methods](#) [Events](#)

Unit

advgrid

TGridCheckBox.FGrid

TGridCheckBox

FGrid: TAdvStringGrid;

TGridCheckBox Events

[TGridCheckBox](#) [Alphabetically](#)

[Legend](#)

Derived from TCheckBox

- [OnClick](#)
- [OnContextPopup](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnStartDock](#)
- [OnStartDrag](#)

Derived from TWinControl

Derived from TControl

TGridCheckBox Events

[TGridCheckBox](#)

[By object](#)

[Legend](#)

- [OnClick](#)
- [OnContextPopup](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnMouseDown](#)
- [OnMouseMove](#)
- [OnMouseUp](#)
- [OnStartDock](#)
- [OnStartDrag](#)

TGridCheckBox Methods

[TGridCheckBox](#) [Alphabetically](#) [Legend](#)

Derived from TCustomCheckBox

[Create](#)
[GetControlsAlignment](#)

Derived from TButtonControl

Derived from TWinControl

[Broadcast](#)
[CanFocus](#)
[ContainsControl](#)
[ControlAtPos](#)
[CreateParented](#)
[DefaultHandler](#)
[Destroy](#)
[DisableAlign](#)
[DockDrop](#)
[EnableAlign](#)
[FindChildControl](#)
[FlipChildren](#)
[Focused](#)
[GetTabOrderList](#)
[HandleAllocated](#)
[HandleNeeded](#)
[InsertControl](#)
[Invalidate](#)
[PaintTo](#)
[Realign](#)
[RemoveControl](#)
[Repaint](#)
[ScaleBy](#)
[ScrollBy](#)
[SetBounds](#)
[SetFocus](#)
[Update](#)
[UpdateControlState](#)

Derived from TControl

[BeginDrag](#)
[BringToFront](#)
[ClientToParent](#)
[ClientToScreen](#)
[Dock](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EndDrag](#)
[GetParentComponent](#)
[GetTextBuf](#)
[GetTextLen](#)
[HasParent](#)
[Hide](#)
[InitiateAction](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[ParentToClient](#)
[Perform](#)

[Refresh](#)
[ReplaceDockedControl](#)
[ScreenToClient](#)
[SendToBack](#)
[SetTextBuf](#)
[Show](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

Derived from TComponent

[BeforeDestruction](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridCheckBox Methods

[TGridCheckBox](#) [By object](#) [Legend](#)

[Assign](#)
[BeforeDestruction](#)
[BeginDrag](#)
[BringToFront](#)
[Broadcast](#)
[CanFocus](#)
[ClientToParent](#)
[ClientToScreen](#)
[ContainsControl](#)
[ControlAtPos](#)
[Create](#)
[CreateParented](#)
[DefaultHandler](#)
[Destroy](#)
[DestroyComponents](#)
[Destroying](#)
[DisableAlign](#)
[Dock](#)
[DockDrop](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EnableAlign](#)
[EndDrag](#)
[ExecuteAction](#)
[FindChildControl](#)
[FindComponent](#)
[FlipChildren](#)
[Focused](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetControlsAlignment](#)
[GetNamePath](#)
[GetParentComponent](#)
[GetTabOrderList](#)
[GetTextBuf](#)
[GetTextLen](#)
[HandleAllocated](#)
[HandleNeeded](#)
[HasParent](#)
[Hide](#)
[InitiateAction](#)
[InsertComponent](#)
[InsertControl](#)
[Invalidate](#)
[IsImplementorOf](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[PaintTo](#)
[ParentToClient](#)
[Perform](#)
[Realign](#)
[ReferenceInterface](#)
[Refresh](#)
[RemoveComponent](#)

[RemoveControl](#)
[RemoveFreeNotification](#)
[Repaint](#)
[ReplaceDockedControl](#)
[SafeCallException](#)
[ScaleBy](#)
[ScreenToClient](#)
[ScrollBy](#)
[SendToBack](#)
[SetBounds](#)
[SetFocus](#)
[SetSubComponent](#)
[SetTextBuf](#)
[Show](#)
[Update](#)
[UpdateAction](#)
[UpdateControlState](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

TGridCheckBox Properties

[TGridCheckBox](#) [Alphabetically](#) [Legend](#)

In TGridCheckBox

[FGrid](#)

Derived from TCheckBox

- [Action](#)
- [Alignment](#)
- [AllowGrayed](#)
- [Anchors](#)
- [BiDiMode](#)
- [Caption](#)
- [Checked](#)
- [Color](#)
- [Constraints](#)
- [Ctl3D](#)
- [DragCursor](#)
- [DragKind](#)
- [DragMode](#)
- [Enabled](#)
- [Font](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentCtl3D](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [PopupMenu](#)
- [ShowHint](#)
- [State](#)
- [TabOrder](#)
- [TabStop](#)
- [Visible](#)

Derived from TButtonControl

Derived from TWinControl

- [AlignDisabled](#)
- [Brush](#)
- [class](#)
- [ControlCount](#)
- [Controls](#)
- [DockClientCount](#)
- [DockClients](#)
- [DoubleBuffered](#)
- [Handle](#)
- [ParentWindow](#)
- [Showing](#)
- [VisibleDockClientCount](#)

Derived from TControl

- [Align](#)
- [BoundsRect](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
- [DockOrientation](#)

- [Floating](#)
- [FloatingDockSiteClass](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LRDockWidth](#)
- [Parent](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [Width](#)
- [WindowProc](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridCheckBox Properties

[TGridCheckBox](#) [By object](#) [Legend](#)

- [Action](#)
- [Align](#)
- [AlignDisabled](#)
- [Alignment](#)
- [AllowGrayed](#)
- [Anchors](#)
- [BiDiMode](#)
- [BoundsRect](#)
- [Brush](#)
- [Caption](#)
- [Checked](#)
- [class](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [Color](#)
- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [Constraints](#)
- [ControlCount](#)
- [Controls](#)
- [ControlState](#)
- [ControlStyle](#)
- [Ctl3D](#)
- [Cursor](#)
- [DesignInfo](#)
- [DockClientCount](#)
- [DockClients](#)
- [DockOrientation](#)
- [DoubleBuffered](#)
- [DragCursor](#)
- [DragKind](#)
- [DragMode](#)
- [Enabled](#)
- [FGrid](#)
- [Floating](#)
- [FloatingDockSiteClass](#)
- [Font](#)
- [Handle](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LRDockWidth](#)
- [Name](#)
- [Owner](#)
- [Parent](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentCtl3D](#)

- [ParentFont](#)
- [ParentShowHint](#)
- [ParentWindow](#)
- [PopupMenu](#)
- [ShowHint](#)
- [Showing](#)
- [State](#)
- [TabOrder](#)
- [TabStop](#)
- [Tag](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [VCLComObject](#)
- [Visible](#)
- [VisibleDockClientCount](#)
- [Width](#)
- [WindowProc](#)

TGridColorEvent

Unit

advgrid

TGridColorEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; AState: [TGridDrawState](#); ABrush: [TBrush](#); AFont: [TFont](#)) **of object**;

TGridCombo

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridCombo.DoChange

[TGridCombo](#)

procedure DoChange;

TGridCombo.SizeDropDownWidth

[TGridCombo](#)

procedure SizeDropDownWidth;

TGridCombo.FGrid

TGridCombo

FGrid: TAdvStringGrid;

TGridCombo Events

[TGridCombo](#)

[Alphabetically](#)

[Legend](#)

TGridCombo Events

[TGridCombo](#)

[.By object](#)

[Legend](#)

TGridCombo Methods

[TGridCombo](#) [Alphabetically](#)

[Legend](#)

In TGridCombo

[DoChange](#)

[SizeDropDownWidth](#)

TGridCombo Methods

[TGridCombo](#)

[.By object](#)

[Legend](#)

[DoChange](#)

[SizeDropDownWidth](#)

TGridCombo Properties

[TGridCombo](#) [Alphabetically](#)

[Legend](#)

In TGridCombo

[FGrid](#)

TGridCombo Properties

[TGridCombo](#)

[.By object](#)

[Legend](#)

[FGrid](#)

TGridDatePicker

[Hierarchy](#) [Properties](#)

[Methods](#) [Events](#)

Unit

advgrid

TGridDatePicker.FGrid

TGridDatePicker

FGrid: TAdvStringGrid;

TGridDatePicker Events

[TGridDatePicker](#) [Alphabetically](#) [Legend](#)

Derived from TDateTimePicker

- [OnChange](#)
- [OnClick](#)
- [OnCloseUp](#)
- [OnContextPopup](#)
- [OnDbClick](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnDropDown](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnStartDock](#)
- [OnStartDrag](#)
- [OnUserInput](#)

Derived from TCommonCalendar

Derived from TWinControl

Derived from TControl

TGridDatePicker Events

[TGridDatePicker](#) [By object](#)

[Legend](#)

- [OnChange](#)
- [OnClick](#)
- [OnCloseUp](#)
- [OnContextPopup](#)
- [OnDbClick](#)
- [OnDragDrop](#)
- [OnDragOver](#)
- [OnDropDown](#)
- [OnEndDock](#)
- [OnEndDrag](#)
- [OnEnter](#)
- [OnExit](#)
- [OnKeyDown](#)
- [OnKeyPress](#)
- [OnKeyUp](#)
- [OnStartDock](#)
- [OnStartDrag](#)
- [OnUserInput](#)

TGridDatePicker Methods

[TGridDatePicker](#) [Alphabetically](#) [Legend](#)

Derived from TDateTimePicker

[Create](#)

Derived from TCommonCalendar

[BoldDays](#)

[Destroy](#)

Derived from TWinControl

[Broadcast](#)

[CanFocus](#)

[ContainsControl](#)

[ControlAtPos](#)

[CreateParented](#)

[DefaultHandler](#)

[DisableAlign](#)

[DockDrop](#)

[EnableAlign](#)

[FindChildControl](#)

[FlipChildren](#)

[Focused](#)

[GetTabOrderList](#)

[HandleAllocated](#)

[HandleNeeded](#)

[InsertControl](#)

[Invalidate](#)

[PaintTo](#)

[Realign](#)

[RemoveControl](#)

[Repaint](#)

[ScaleBy](#)

[ScrollBy](#)

[SetBounds](#)

[SetFocus](#)

[Update](#)

[UpdateControlState](#)

Derived from TControl

[BeginDrag](#)

[BringToFront](#)

[ClientToParent](#)

[ClientToScreen](#)

[Dock](#)

[DragDrop](#)

[Dragging](#)

[DrawTextBiDiModeFlags](#)

[DrawTextBiDiModeFlagsReadingOn](#)

[EndDrag](#)

[GetControlsAlignment](#)

[GetParentComponent](#)

[GetTextBuf](#)

[GetTextLen](#)

[HasParent](#)

[Hide](#)

[InitiateAction](#)

[IsRightToLeft](#)

[ManualDock](#)

[ManualFloat](#)

[MouseWheelHandler](#)

[ParentToClient](#)

[Perform](#)
[Refresh](#)
[ReplaceDockedControl](#)
[ScreenToClient](#)
[SendToBack](#)
[SetTextBuf](#)
[Show](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

Derived from TComponent

[BeforeDestruction](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridDatePicker Methods

[TGridDatePicker](#) [By object](#) [Legend](#)

[Assign](#)
[BeforeDestruction](#)
[BeginDrag](#)
[BoldDays](#)
[BringToFront](#)
[Broadcast](#)
[CanFocus](#)
[ClientToParent](#)
[ClientToScreen](#)
[ContainsControl](#)
[ControlAtPos](#)
[Create](#)
[CreateParented](#)
[DefaultHandler](#)
[Destroy](#)
[DestroyComponents](#)
[Destroying](#)
[DisableAlign](#)
[Dock](#)
[DockDrop](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EnableAlign](#)
[EndDrag](#)
[ExecuteAction](#)
[FindChildControl](#)
[FindComponent](#)
[FlipChildren](#)
[Focused](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetControlsAlignment](#)
[GetNamePath](#)
[GetParentComponent](#)
[GetTabOrderList](#)
[GetTextBuf](#)
[GetTextLen](#)
[HandleAllocated](#)
[HandleNeeded](#)
[HasParent](#)
[Hide](#)
[InitiateAction](#)
[InsertComponent](#)
[InsertControl](#)
[Invalidate](#)
[IsImplementorOf](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[PaintTo](#)
[ParentToClient](#)
[Perform](#)
[Realign](#)
[ReferenceInterface](#)
[Refresh](#)

[RemoveComponent](#)
[RemoveControl](#)
[RemoveFreeNotification](#)
[Repaint](#)
[ReplaceDockedControl](#)
[SafeCallException](#)
[ScaleBy](#)
[ScreenToClient](#)
[ScrollBy](#)
[SendToBack](#)
[SetBounds](#)
[SetFocus](#)
[SetSubComponent](#)
[SetTextBuf](#)
[Show](#)
[Update](#)
[UpdateAction](#)
[UpdateControlState](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

TGridDatePicker Properties

[TGridDatePicker](#) [Alphabetically](#) [Legend](#)

In TGridDatePicker

[FGrid](#)

Derived from TDateTimePicker

- [Anchors](#)
- [BevelEdges](#)
- [BevelInner](#)
- [BevelKind](#)
- [BevelOuter](#)
- [BevelWidth](#)
- [BiDiMode](#)
- [CalAlignment](#)
- [CalColors](#)
- [Checked](#)
- [Color](#)
- [Constraints](#)
- [Date](#)
- [DateFormat](#)
- [DateMode](#)
- [DateTime](#)
- [DragCursor](#)
- [DragKind](#)
- [DragMode](#)
- [DroppedDown](#)
- [Enabled](#)
- [Font](#)
- [Format](#)
- [ImeMode](#)
- [ImeName](#)
- [Kind](#)
- [MaxDate](#)
- [MinDate](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [ParseInput](#)
- [PopupMenu](#)
- [ShowCheckbox](#)
- [ShowHint](#)
- [TabOrder](#)
- [TabStop](#)
- [Time](#)
- [Visible](#)

Derived from TCommonCalendar

Derived from TWinControl

- [AlignDisabled](#)
- [Brush](#)
- [class](#)
- [ControlCount](#)
- [Controls](#)
- [DockClientCount](#)
- [DockClients](#)
- [DoubleBuffered](#)
- [Handle](#)
- [ParentWindow](#)
- [Showing](#)

- [VisibleDockClientCount](#)

Derived from TControl

- [Action](#)
- [Align](#)
- [BoundsRect](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
- [DockOrientation](#)
- [Floating](#)
- [FloatingDockSiteClass](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LRDockWidth](#)
- [Parent](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [Width](#)
- [WindowProc](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridDatePicker Properties

[TGridDatePicker](#) [By object](#) [Legend](#)

[Action](#)

[Align](#)

- [AlignDisabled](#)
- [Anchors](#)
- [BevelEdges](#)
- [BevelInner](#)
- [BevelKind](#)
- [BevelOuter](#)
- [BevelWidth](#)
- [BiDiMode](#)
- [BoundsRect](#)
- [Brush](#)
- [CalAlignment](#)
- [CalColors](#)
- [Checked](#)
- [class](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [Color](#)
- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [Constraints](#)
- [ControlCount](#)
- [Controls](#)
- [ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
- [Date](#)
- [DateFormat](#)
- [DateMode](#)
- [DateTime](#)
- [DesignInfo](#)
- [DockClientCount](#)
- [DockClients](#)
- [DockOrientation](#)

[DoubleBuffered](#)

- [DragCursor](#)
- [DragKind](#)
- [DragMode](#)
- [DroppedDown](#)
- [Enabled](#)

[FGrid](#)

- [Floating](#)
- [FloatingDockSiteClass](#)
- [Font](#)
- [Format](#)
- [Handle](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)

- [HostDockSite](#)
- [ImeMode](#)
- [ImeName](#)
- [Kind](#)
- [Left](#)
- [LRDockWidth](#)
- [MaxDate](#)
- [MinDate](#)
- [Name](#)
- [Owner](#)
- [Parent](#)
- [ParentBiDiMode](#)
- [ParentColor](#)
- [ParentFont](#)
- [ParentShowHint](#)
- [ParentWindow](#)
- [ParseInput](#)
- [PopupMenu](#)
- [ShowCheckbox](#)
- [ShowHint](#)
- [Showing](#)
- [TabOrder](#)
- [TabStop](#)
- [Tag](#)
- [TBDockHeight](#)
- [Time](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [VCLComObject](#)
- [Visible](#)
- [VisibleDockClientCount](#)
- [Width](#)
- [WindowProc](#)

TGridDropSource

[Hierarchy](#) [Properties](#)

[Methods](#)

Unit

advgrid

TGridDropSource.CurrentEffect

[TGridDropSource](#)

procedure CurrentEffect(dwEffect: **Longint**);

TGridDropSource.QueryDrag

[TGridDropSource](#)

```
procedure QueryDrag;
```

TGridDropSource.LastEffect

[TGridDropSource](#)

property LastEffect: Integer;

TGridDropSource Events

[TGridDropSource](#) [Alphabetically](#)

[Legend](#)

TGridDropSource Events

[TGridDropSource](#) [.By object](#)

[Legend](#)

TGridDropSource Methods

[TGridDropSource](#) [Alphabetically](#) [Legend](#)

In TGridDropSource

[CurrentEffect](#)

[QueryDrag](#)

TGridDropSource Methods

[TGridDropSource](#) [By object](#) [Legend](#)

[CurrentEffect](#)
[QueryDrag](#)

TGridDropSource Properties

[TGridDropSource](#) [Alphabetically](#) [Legend](#)

In TGridDropSource

■

■ [LastEffect](#)

TGridDropSource Properties

[TGridDropSource](#) [_By object](#) [Legend](#)

-
- [LastEffect](#)

TGridDropTarget

[Hierarchy](#) [Methods](#)

Unit

advgrid

TGridDropTarget.DragMouseLeave

[TGridDropTarget](#)

```
procedure DragMouseLeave;
```


TGridDropTarget.DragMouseMove

[TGridDropTarget](#)

procedure DragMouseMove(pt: TPoint; **var** Allow: Boolean; DropFormats: TDropFormats);

TGridDropTarget.DropCol

[TGridDropTarget](#)

procedure DropCol(pt: TPoint; Col: **Integer**);

TGridDropTarget.DropFiles

[TGridDropTarget](#)

procedure DropFiles(pt: TPoint; files: [tstrings](#));

TGridDropTarget.DropRTF

[TGridDropTarget](#)

```
procedure DropRTF(pt: TPoint; s: string);
```

TGridDropTarget.DropText

[TGridDropTarget](#)

```
procedure DropText(pt: TPoint; s: string);
```

TGridDropTarget Events

[TGridDropTarget](#)

[Alphabetically](#)

[Legend](#)

TGridDropTarget Events

[TGridDropTarget](#)

[By object](#)

[Legend](#)

TGridDropTarget Methods

[TGridDropTarget](#) [Alphabetically](#) [Legend](#)

In TGridDropTarget

[DragMouseLeave](#)

[DragMouseMove](#)

[DropCol](#)

[DropFiles](#)

[DropRTF](#)

[DropText](#)

TGridDropTarget Methods

TGridDropTarget By object Legend

DragMouseLeave

DragMouseMove

DropCol

DropFiles

DropRTF

DropText

TGridDropTarget Properties

[TGridDropTarget](#) [Alphabetically](#) [Legend](#)

TGridDropTarget Properties

[TGridDropTarget](#)

[By object](#)

[Legend](#)

TGridEditBtn

[Hierarchy](#) [Methods](#)

Unit

advgrid

TGridEditBtn Events

[TGridEditBtn](#)

[Alphabetically](#)

[Legend](#)

TGridEditBtn Events

[TGridEditBtn](#)

[.By object](#)

[Legend](#)

TGridEditBtn Methods

[TGridEditBtn](#)

[Alphabetically](#)

[Legend](#)

TGridEditBtn Methods

[TGridEditBtn](#)

[.By object](#)

[Legend](#)

TGridEditBtn Properties

[TGridEditBtn](#)

[Alphabetically](#)

[Legend](#)

TGridEditBtn Properties

[TGridEditBtn](#)

[.By object](#)

[Legend](#)

TGridFilePicIO

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridFilePicIO.Picture

[TGridFilePicIO](#)

property Picture: TFilePicture;

TGridFilePicIO Events

[TGridFilePicIO](#)

[Alphabetically](#)

[Legend](#)

TGridFilePicIO Events

[TGridFilePicIO](#)

[.By object](#)

[Legend](#)

TGridFilePicIO Methods

[TGridFilePicIO](#) [Alphabetically](#)

[Legend](#)

Derived from TComponent

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridFilePicIO Methods

[TGridFilePicIO](#)

[By object](#)

[Legend](#)

[Assign](#)

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

TGridFilePicIO Properties

[TGridFilePicIO](#)

[Alphabetically](#)

[Legend](#)

In TGridFilePicIO

- [Picture](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridFilePicIO Properties

TGridFilePicIO

.By object

Legend

- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- DesignInfo
- Name
- Owner
- Picture
- Tag
- VCLComObject

TGridFormatEvent

Unit

advgrid

TGridFormatEvent=**procedure**(Sender: TObject; ACol: **Integer**; **var** AStyle: [TSortStyle](#); **var** aPrefix, aSuffix: **string**) **of object**;

TGridGraphicIO

[Hierarchy](#) [Properties](#)

[Methods](#)

Unit

advgrid

TGridGraphicIO.CellGraphic

[TGridGraphicIO](#)

property CellGraphic: [TCellGraphic](#);

TGridGraphicIO Events

[TGridGraphicIO](#)

[Alphabetically](#)

[Legend](#)

TGridGraphicIO Events

[TGridGraphicIO](#)

[.By object](#)

[Legend](#)

TGridGraphicIO Methods

[TGridGraphicIO](#) [Alphabetically](#)

[Legend](#)

Derived from TComponent

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridGraphicIO Methods

[TGridGraphicIO](#)

[By object](#)

[Legend](#)

[Assign](#)

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

TGridGraphicIO Properties

[TGridGraphicIO](#) [Alphabetically](#) [Legend](#)

In TGridGraphicIO

- [CellGraphic](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridGraphicIO Properties

[TGridGraphicIO](#)

[By object](#)

[Legend](#)

- [CellGraphic](#)
- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridHintEvent

Unit

advgrid

TGridHintEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; var hintstr: **string**) **of object**;

TGridConIO

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridconIO.Icon

[TGridconIO](#)

property Icon: [TIcon](#);

TGridIconIO Events

[TGridIconIO](#)

[Alphabetically](#)

[Legend](#)

TGridIconIO Events

[TGridIconIO](#)

[.By object](#)

[Legend](#)

TGridIconIO Methods

[TGridIconIO](#) [Alphabetically](#)

[Legend](#)

Derived from TComponent

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridIconIO Methods

TGridIconIO

By object

Legend

Assign

BeforeDestruction

Create

Destroy

DestroyComponents

Destroying

ExecuteAction

FindComponent

FreeNotification

FreeOnRelease

GetNamePath

GetParentComponent

HasParent

InsertComponent

IsImplementorOf

ReferenceInterface

RemoveComponent

RemoveFreeNotification

SafeCallException

SetSubComponent

UpdateAction

TGridIconO Properties

[TGridIconO](#) [Alphabetically](#)

[Legend](#)

In TGridIconO

- [Icon](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridIconO Properties

TGridIconO

By object

Legend

- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- DesignInfo
- Icon
- Name
- Owner
- Tag
- VCLComObject

TGridItem

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridItem.Idx

[TGridItem](#)

property Idx: Integer;

TGridItem.Items

[TGridItem](#)

property Items: [tstrings](#);

TGridItem Events

[TGridItemAlphabetically](#)

[Legend](#)

TGridItem Events

[TGridItem](#) [By object](#)

[Legend](#)

TGridItem Methods

[TGridItemAlphabetically](#) [Legend](#)

Derived from TCollectionItem

[Create](#)

[Destroy](#)

[GetNamePath](#)

Derived from TPersistent

[Assign](#)

TGridItem Methods

[TGridItem](#) [By object](#)

[Legend](#)

[Assign](#)

[Create](#)

[Destroy](#)

[GetNamePath](#)

TGridItem Properties

[TGridItemAlphabetically](#)

[Legend](#)

In TGridItem

- [Idx](#)
- [Items](#)

Derived from TCollectionItem

- [Collection](#)
- [DisplayName](#)
- [ID](#)
- [Index](#)

TGridItem Properties

TGridItem.By object

Legend

Collection

DisplayName

- ID
- Idx
- Index
- Items

TGridLook

Unit

advgrid

TGridLook=(glStandard, glSoft, glClassic, glTMS); **Description**

Value	Meaning
glStandard	Standard Window grid look
glSoft	Soft flat grid look
glTMS	TMS style with shaded headers
glClassic	Classic heavy 3D style

TGridPicIO

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridPicO.Picture

[TGridPicO](#)

property Picture: [TPicture](#);

TGridPicIO Events

[TGridPicIO](#)

[Alphabetically](#)

[Legend](#)

TGridPicIO Events

[TGridPicIO](#)

[.By object](#)

[Legend](#)

TGridPicIO Methods

[TGridPicIO](#)

[Alphabetically](#)

[Legend](#)

Derived from TComponent

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridPicIO Methods

TGridPicIO

By object

Legend

Assign
BeforeDestruction
Create
Destroy
DestroyComponents
Destroying
ExecuteAction
FindComponent
FreeNotification
FreeOnRelease
GetNamePath
GetParentComponent
HasParent
InsertComponent
IsImplementorOf
ReferenceInterface
RemoveComponent
RemoveFreeNotification
SafeCallException
SetSubComponent
UpdateAction

TGridPicIO Properties

[TGridPicIO](#)

[Alphabetically](#)

[Legend](#)

In TGridPicIO

- [Picture](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridPicIO Properties

TGridPicIO

.By object

Legend

- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- DesignInfo
- Name
- Owner
- Picture
- Tag
- VCLComObject

TGridPrintColumnWidthEvent

Unit

advgrid

TGridPrintColumnWidthEvent=**procedure**(Sender: TObject; ACol: **Integer**; var Width: **Integer**) **of object**;

TGridPrintNewPageEvent

Unit

advgrid

TGridPrintNewPageEvent=**procedure**(Sender: TObject; ARow: **Integer**; **var** NewPage: Boolean) **of**
object;

TGridPrintPageDoneEvent

Unit

advgrid

TGridPrintPageDoneEvent=**procedure**(Sender: TObject; Canvas: [TCanvas](#); LastRow, LastRowOffset, LastPage, PageXSize, PageYSize: **Integer**) **of object**;

TGridPrintPageEvent

Unit

advgrid

TGridPrintPageEvent=**procedure**(Sender: TObject; Canvas: [TCanvas](#); PageNr, PageXSize, PageYSize: **Integer**) of object;

TGridPrintRowHeightEvent

Unit

advgrid

TGridPrintRowHeightEvent=**procedure**(Sender: TObject; ARow: **Integer**; **var** Height: **Integer**) of object;

TGridPrintStartEvent

Unit

advgrid

TGridPrintStartEvent=**procedure**(Sender: TObject; NrOfPages: **Integer**; **var** FromPage, ToPage: **Integer**) of object;

TGridProgressEvent

Unit

advgrid

TGridProgressEvent=**procedure**(Sender: TObject; progress: **smallint**) of object;

TGridPropIO

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridPropIO.ColCount

[TGridPropIO](#)

property ColCount: **Integer**;

TGridPropIO.ColWidths

[TGridPropIO](#)

property ColWidths: **string**;

TGridPropIO.FullGrid

[TGridPropIO](#)

property FullGrid: Boolean;

TGridPropIO.RowCount

[TGridPropIO](#)

property RowCount: **Integer**;

TGridPropIO.RowHeights

[TGridPropIO](#)

property RowHeights: **string**;

TGridPropIO Events

[TGridPropIO](#)

[Alphabetically](#)

[Legend](#)

TGridPropIO Events

TGridPropIO

.By object

Legend

TGridPropIO Methods

[TGridPropIO](#) [Alphabetically](#)

[Legend](#)

Derived from TComponent

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridPropIO Methods

TGridPropIO

By object

Legend

Assign

BeforeDestruction

Create

Destroy

DestroyComponents

Destroying

ExecuteAction

FindComponent

FreeNotification

FreeOnRelease

GetNamePath

GetParentComponent

HasParent

InsertComponent

IsImplementorOf

ReferenceInterface

RemoveComponent

RemoveFreeNotification

SafeCallException

SetSubComponent

UpdateAction

TGridPropIO Properties

[TGridPropIO](#) [Alphabetically](#)

[Legend](#)

In TGridPropIO

- [ColCount](#)
- [ColWidths](#)
- [FullGrid](#)
- [RowCount](#)
- [RowHeights](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridPropIO Properties

TGridPropIO

By object

Legend

- ColCount
- ColWidths
- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- DesignInfo
- FullGrid
- Name
- Owner
- RowCount
- RowHeights
- Tag
- VCLComObject

TGridSLIO

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridSLIO.Strings

TGridSLIO

property Strings: TStringList;

TGridSLIO Events

[TGridSLIO](#)

[Alphabetically](#)

[Legend](#)

TGridSLIO Events

TGridSLIO

.By object

Legend

TGridSLIO Methods

[TGridSLIO](#)

[Alphabetically](#)

[Legend](#)

Derived from TComponent

[BeforeDestruction](#)

[Create](#)

[Destroy](#)

[DestroyComponents](#)

[Destroying](#)

[ExecuteAction](#)

[FindComponent](#)

[FreeNotification](#)

[FreeOnRelease](#)

[GetNamePath](#)

[GetParentComponent](#)

[HasParent](#)

[InsertComponent](#)

[IsImplementorOf](#)

[ReferenceInterface](#)

[RemoveComponent](#)

[RemoveFreeNotification](#)

[SafeCallException](#)

[SetSubComponent](#)

[UpdateAction](#)

Derived from TPersistent

[Assign](#)

TGridSLIO Methods

TGridSLIO

By object

Legend

Assign

BeforeDestruction

Create

Destroy

DestroyComponents

Destroying

ExecuteAction

FindComponent

FreeNotification

FreeOnRelease

GetNamePath

GetParentComponent

HasParent

InsertComponent

IsImplementorOf

ReferenceInterface

RemoveComponent

RemoveFreeNotification

SafeCallException

SetSubComponent

UpdateAction

TGridSLIO Properties

[TGridSLIO](#)

[Alphabetically](#)

[Legend](#)

In TGridSLIO

- [Strings](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

TGridSLIO Properties

TGridSLIO

By object

Legend

- ComObject
- ComponentCount
- ComponentIndex
- Components
- ComponentState
- ComponentStyle
- DesignInfo
- Name
- Owner
- Strings
- Tag
- VCLComObject

TGridSpin

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

TGridSpin.FGrid

TGridSpin

FGrid: TAdvStringGrid;

TGridSpin Events

[TGridSpin](#)

[Alphabetically](#)

[Legend](#)

TGridSpin Events

TGridSpin

.By object

Legend

TGridSpin Methods

[TGridSpin](#)

[Alphabetically](#)

[Legend](#)

TGridSpin Methods

TGridSpin

.By object

Legend

TGridSpin Properties

[TGridSpin](#)

[Alphabetically](#)

[Legend](#)

In TGridSpin

[FGrid](#)

TGridSpin Properties

[TGridSpin](#)

[.By object](#)

[Legend](#)

[FGrid](#)

TGridUnitEditBtn

[Hierarchy](#) [Methods](#)

Unit

advgrid

TGridUnitEditBtn Events

[TGridUnitEditBtn](#)

[Alphabetically](#)

[Legend](#)

TGridUnitEditBtn Events

[TGridUnitEditBtn](#)

[.By object](#)

[Legend](#)

TGridUnitEditBtn Methods

[TGridUnitEditBtn](#) [Alphabetically](#) [Legend](#)

TGridUnitEditBtn Methods

[TGridUnitEditBtn](#)

[.By object](#)

[Legend](#)

TGridUnitEditBtn Properties

[TGridUnitEditBtn](#) [Alphabetically](#) [Legend](#)

TGridUnitEditBtn Properties

[TGridUnitEditBtn](#)

[.By object](#)

[Legend](#)

THomeEndAction

Unit

advgrid

THomeEndAction=(heFirstLastColumn, heFirstLastRow); **Description**

Value

Meaning

heFirstLastColumn Move to first / last column with Home / End key

heFirstLastRow Move to first / last row with Home / End key

THTMLHintWindow

[Hierarchy](#)

[Properties](#)

[Methods](#) [Events](#)

Unit

advgrid

THTMLHintWindow.ActivateHint

[THTMLHintWindow](#)

```
procedure ActivateHint(Rect: TRect; const AHint: string);
```

THTMLHintWindow.CalcHintRect

[THTMLHintWindow](#)

```
function CalcHintRect(MaxWidth: Integer; const AHint: string; AData: Pointer): TRect;
```

[THTMLHintWindow Events](#)

[THTMLHintWindow](#) [Alphabetically](#)

[Legend](#)

Derived from TWinControl

Derived from TControl

THTMLHintWindow Events

[THTMLHintWindow](#) [By object](#)

[Legend](#)

THTMLHintWindow Methods

[THTMLHintWindow](#) [Alphabetically](#) [Legend](#)

In THTMLHintWindow

[ActivateHint](#)
[CalcHintRect](#)

Derived from THintWindow

[ActivateHintData](#)
[Create](#)
[IsHintMsg](#)
[ReleaseHandle](#)

Derived from TCustomControl

[Destroy](#)

Derived from TWinControl

[Broadcast](#)
[CanFocus](#)
[ContainsControl](#)
[ControlAtPos](#)
[CreateParented](#)
[DefaultHandler](#)
[DisableAlign](#)
[DockDrop](#)
[EnableAlign](#)
[FindChildControl](#)
[FlipChildren](#)
[Focused](#)
[GetTabOrderList](#)
[HandleAllocated](#)
[HandleNeeded](#)
[InsertControl](#)
[Invalidate](#)
[PaintTo](#)
[Realign](#)
[RemoveControl](#)
[Repaint](#)
[ScaleBy](#)
[ScrollBy](#)
[SetBounds](#)
[SetFocus](#)
[Update](#)
[UpdateControlState](#)

Derived from TControl

[BeginDrag](#)
[BringToFront](#)
[ClientToParent](#)
[ClientToScreen](#)
[Dock](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EndDrag](#)
[GetControlsAlignment](#)
[GetParentComponent](#)
[GetTextBuf](#)
[GetTextLen](#)
[HasParent](#)
[Hide](#)

[InitiateAction](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[ParentToClient](#)
[Perform](#)
[Refresh](#)
[ReplaceDockedControl](#)
[ScreenToClient](#)
[SendToBack](#)
[SetTextBuf](#)
[Show](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

Derived from TComponent

[BeforeDestruction](#)
[DestroyComponents](#)
[Destroying](#)
[ExecuteAction](#)
[FindComponent](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetNamePath](#)
[InsertComponent](#)
[IsImplementorOf](#)
[ReferenceInterface](#)
[RemoveComponent](#)
[RemoveFreeNotification](#)
[SafeCallException](#)
[SetSubComponent](#)
[UpdateAction](#)

Derived from TPersistent

[Assign](#)

THTMLHintWindow Methods

[THTMLHintWindow](#) [By object](#)

[Legend](#)

[ActivateHint](#)
[ActivateHintData](#)
[Assign](#)
[BeforeDestruction](#)
[BeginDrag](#)
[BringToFront](#)
[Broadcast](#)
[CalcHintRect](#)
[CanFocus](#)
[ClientToParent](#)
[ClientToScreen](#)
[ContainsControl](#)
[ControlAtPos](#)
[Create](#)
[CreateParented](#)
[DefaultHandler](#)
[Destroy](#)
[DestroyComponents](#)
[Destroying](#)
[DisableAlign](#)
[Dock](#)
[DockDrop](#)
[DragDrop](#)
[Dragging](#)
[DrawTextBiDiModeFlags](#)
[DrawTextBiDiModeFlagsReadingOn](#)
[EnableAlign](#)
[EndDrag](#)
[ExecuteAction](#)
[FindChildControl](#)
[FindComponent](#)
[FlipChildren](#)
[Focused](#)
[FreeNotification](#)
[FreeOnRelease](#)
[GetControlsAlignment](#)
[GetNamePath](#)
[GetParentComponent](#)
[GetTabOrderList](#)
[GetTextBuf](#)
[GetTextLen](#)
[HandleAllocated](#)
[HandleNeeded](#)
[HasParent](#)
[Hide](#)
[InitiateAction](#)
[InsertComponent](#)
[InsertControl](#)
[Invalidate](#)
[IsHintMsg](#)
[IsImplementorOf](#)
[IsRightToLeft](#)
[ManualDock](#)
[ManualFloat](#)
[MouseWheelHandler](#)
[PaintTo](#)
[ParentToClient](#)
[Perform](#)

[Realign](#)
[ReferenceInterface](#)
[Refresh](#)
[ReleaseHandle](#)
[RemoveComponent](#)
[RemoveControl](#)
[RemoveFreeNotification](#)
[Repaint](#)
[ReplaceDockedControl](#)
[SafeCallException](#)
[ScaleBy](#)
[ScreenToClient](#)
[ScrollBy](#)
[SendToBack](#)
[SetBounds](#)
[SetFocus](#)
[SetSubComponent](#)
[SetTextBuf](#)
[Show](#)
[Update](#)
[UpdateAction](#)
[UpdateControlState](#)
[UseRightToLeftAlignment](#)
[UseRightToLeftReading](#)
[UseRightToLeftScrollBar](#)

THTMLHintWindow Properties

[THTMLHintWindow](#) [Alphabetically](#) [Legend](#)

Derived from THintWindow

[BiDiMode](#)
[Canvas](#)
[Caption](#)
[Color](#)
[Font](#)

Derived from TWinControl

- [AlignDisabled](#)
- [Brush](#)
- [class](#)
- [ControlCount](#)
- [Controls](#)
- [DockClientCount](#)
- [DockClients](#)
- [DoubleBuffered](#)
- [Handle](#)
- [ParentWindow](#)
- [Showing](#)
- [TabOrder](#)
- [TabStop](#)
- [VisibleDockClientCount](#)

Derived from TControl

- [Action](#)
- [Align](#)
- [Anchors](#)
- [BoundsRect](#)
- [ClientHeight](#)
- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [Constraints](#)
- [ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
- [DockOrientation](#)
- [Enabled](#)
- [Floating](#)
- [FloatingDockSiteClass](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LRDockWidth](#)
- [Parent](#)
- [ShowHint](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)
- [Visible](#)
- [Width](#)
- [WindowProc](#)

Derived from TComponent

- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [DesignInfo](#)
- [Name](#)
- [Owner](#)
- [Tag](#)
- [VCLComObject](#)

THTMLHintWindow Properties

[THTMLHintWindow](#) [By object](#)

[Legend](#)

[Action](#)

[Align](#)

- [AlignDisabled](#)
- [Anchors](#)
- [BiDiMode](#)
- [BoundsRect](#)
- [Brush](#)
- [Canvas](#)

[Caption](#)

[class](#)

[ClientHeight](#)

- [ClientOrigin](#)
- [ClientRect](#)
- [ClientWidth](#)
- [Color](#)
- [ComObject](#)
- [ComponentCount](#)
- [ComponentIndex](#)
- [Components](#)
- [ComponentState](#)
- [ComponentStyle](#)
- [Constraints](#)
- [ControlCount](#)
- [Controls](#)
- [ControlState](#)
- [ControlStyle](#)
- [Cursor](#)
- [DesignInfo](#)
- [DockClientCount](#)
- [DockClients](#)
- [DockOrientation](#)
- [DoubleBuffered](#)
- [Enabled](#)
- [Floating](#)
- [FloatingDockSiteClass](#)
- [Font](#)
- [Handle](#)
- [Height](#)
- [HelpContext](#)
- [HelpKeyword](#)
- [HelpType](#)
- [Hint](#)
- [HostDockSite](#)
- [Left](#)
- [LRDockWidth](#)
- [Name](#)
- [Owner](#)
- [Parent](#)
- [ParentWindow](#)
- [ShowHint](#)
- [Showing](#)

[TabOrder](#)

[TabStop](#)

- [Tag](#)
- [TBDockHeight](#)
- [Top](#)
- [UndockHeight](#)
- [UndockWidth](#)

VCLComObject

Visible

■ VisibleDockClientCount

■ Width

WindowProc

THTMLSettings

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Controls the export to a HTML file

THTMLSettings.BorderSize

[THTMLSettings](#)

property BorderSize: **Integer default 1; Description**
Sets the table border size in HTML

THTMLSettings.CellPadding

[THTMLSettings](#)

property CellPadding: **Integer default 0**; **Description**
Sets the table cellpadding in HTML

THTMLSettings.CellSpacing

[THTMLSettings](#)

property CellSpacing: **Integer default 0**; **Description**
Sets the table cellspacing in HTML

THTMLSettings.ColWidths

[THTMLSettings](#)

property ColWidths: TIntList;

THTMLSettings.FooterFile

[THTMLSettings](#)

property FooterFile: **string**; **Description**

Sets the file that is optionally appended at the end of the HTML for the grid content.

THTMLSettings.HeaderFile

[THTMLSettings](#)

property HeaderFile: **string**; **Description**

Sets the file that is optionally included before the grid content.

THTMLSettings.PrefixTag

[THTMLSettings](#)

property PrefixTag: **string**; **Description**

Sets the optional HTML tags included before the grid content HTML code

THTMLSettings.SaveColor

[THTMLSettings](#)

property SaveColor: Boolean **default** True; **Description**
When true, saves color information in HTML

THTMLSettings.SaveFonts

[THTMLSettings](#)

property SaveFonts: Boolean **default** True; **Description**
When true, saves font information in HTML

THTMLSettings.SuffixTag

[THTMLSettings](#)

property SuffixTag: **string**; **Description**

Sets the optional HTML tags included after the grid content HTML code

THTMLSettings.TableStyle

[THTMLSettings](#)

property TableStyle: **string**; **Description**

Sets optional table style tags in <TABLE> specifier in HTML

THTMLSettings.Width

[THTMLSettings](#)

property Width: **Integer**; **Description**

Sets table width in %

THTMLSettings.XHTML

[THTMLSettings](#)

property XHTML: Boolean;

THTMLSettings Events

[THTMLSettings](#)

[Alphabetically](#)

[Legend](#)

THTMLSettings Events

[THTMLSettings](#)

[By object](#)

[Legend](#)

[THTMLSettings Methods](#)

[THTMLSettings](#) [Alphabetically](#)

[Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

THTMLSettings Methods

[THTMLSettings](#)

[By object](#)

[Legend](#)

[Assign](#)

[Destroy](#)

[GetNamePath](#)

THTMLSettings Properties

[THTMLSettings](#) [Alphabetically](#) [Legend](#)

In THTMLSettings

- [BorderSize](#)
- [CellPadding](#)
- [CellSpacing](#)
- [ColWidths](#)
- [FooterFile](#)
- [HeaderFile](#)
- [PrefixTag](#)
- [SaveColor](#)
- [SaveFonts](#)
- [SuffixTag](#)
- [TableStyle](#)
- [Width](#)
- [XHTML](#)

THTMLSettings Properties

[THTMLSettings](#)

[By object](#)

[Legend](#)

- [BorderSize](#)
- [CellPadding](#)
- [CellSpacing](#)
- [ColWidths](#)
- [FooterFile](#)
- [HeaderFile](#)
- [PrefixTag](#)
- [SaveColor](#)
- [SaveFonts](#)
- [SuffixTag](#)
- [TableStyle](#)
- [Width](#)
- [XHTML](#)

TInsertPosition

Unit

advgrid

TInsertPosition=(pInsertBefore, pInsertAfter); **Description**

Value	Meaning
--------------	----------------

pInsertBefore	Insert row before currently selected row
---------------	--

pInsertAfter	Insert row after currently selected row
--------------	---

TIntelliPan

Unit

advgrid

TIntelliPan=(ipVertical, ipHorizontal, ipBoth, ipNone); **Description**

Value **Meaning**

ipVertical Panning with wheel mouse in vertical direction only

ipHorizontal Panning with wheel mouse in horizontal direction only

ipBoth Panning with wheel mouse in both directions

ipNone No panning with wheel mouse in grid

TIsFixedCellEvent

Unit

advgrid

TIsFixedCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; **var** IsFixed: Boolean) **of object**;

TIsPasswordCellEvent

Unit

advgrid

TIsPasswordCellEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; **var** IsPassword: Boolean) **of**
object;

TMouseActions

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Holds all settings of enhanced mouse handling in the grid

TMouseActions.AllColumnSize

[TMouseActions](#)

property AllColumnSize: Boolean **default** False; **Description**
Sizes all columns proportionally when a column size changes

TMouseActions.AllRowSize

[TMouseActions](#)

property AllRowSize: Boolean **default** False; **Description**
Sizes all rows proportionally when the a row size changes

TMouseActions.AllSelect

[TMouseActions](#)

property AllSelect: Boolean **default** False; **Description**

Allows mouse to select all cells when clicking top left fixed cell.

TMouseActions.CaretPositioning

[TMouseActions](#)

property CaretPositioning: Boolean **default** False; **Description**

When true, immediately puts caret at right position in the text when editing.

TMouseActions.ColSelect

[TMouseActions](#)

property ColSelect: Boolean **default** False; **Description**

When true, allows mouse to select column when clicking top fixed cell.

TMouseActions.DirectComboDrop

[TMouseActions](#)

property DirectComboDrop: Boolean **default** False; **Description**

When true, the combobox immediately drops down when the cell is clicked.

TMouseActions.DirectEdit

[TMouseActions](#)

property DirectEdit: Boolean **default** False; **Description**
When true, immediately shows editor when clicking a cell.

TMouseActions.DisjunctCellSelect

[TMouseActions](#) [See also](#)

property DisjunctCellSelect: Boolean **default** False; **Description**
When true, disjunct cell selection is possible.

TMouseActions.DisjunctRowSelect

[TMouseActions](#)

property DisjunctRowSelect: Boolean **default** False; **Description**
Enables or disables disjunct row selection.

TMouseActions.RangeSelectAndEdit

[TMouseActions](#)

property RangeSelectAndEdit: Boolean **default** False; **Description**

When true, range selection and single cell editing can be combined. Single cell editing is started by mouse double click on a cell.

TMouseActions.RowSelect

[TMouseActions](#)

property RowSelect: Boolean **default** False; **Description**

When true, allows mouse to select row when clicking left fixed cell.

TMouseActions.SizeFixedCol

[TMouseActions](#)

property SizeFixedCol: Boolean **default** False; **Description**

When true the first fixed column can be sized at run-time over the full height of the grid.

TMouseActions Events

[TMouseActions](#)

[Alphabetically](#)

[Legend](#)

TMouseActions Events

[TMouseActions](#)

[By object](#)

[Legend](#)

TMouseActions Methods

[TMouseActions](#)

[Alphabetically](#)

[Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TMouseActions Methods

[TMouseActions](#)

[_By object](#)

[Legend](#)

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TMouseActions Properties

[TMouseActions](#) [Alphabetically](#) [Legend](#)

In TMouseActions

- [AllColumnSize](#)
- [AllRowSize](#)
- [AllSelect](#)
- [CaretPositioning](#)
- [ColSelect](#)
- [DirectComboDrop](#)
- [DirectEdit](#)
- [DisjunctCellSelect](#)
- [DisjunctRowSelect](#)
- [RangeSelectAndEdit](#)
- [RowSelect](#)
- [SizeFixedCol](#)

TMouseActions Properties

TMouseActions

By object

Legend

- AllColumnSize
- AllRowSize
- AllSelect
- CaretPositioning
- ColSelect
- DirectComboDrop
- DirectEdit
- DisjunctCellSelect
- DisjunctRowSelect
- RangeSelectAndEdit
- RowSelect
- SizeFixedCol

TMouseSelectMode

Unit

advgrid

TMouseSelectMode=(msNormal, msColumn, msRow, msAll, msURL, msResize); **Description**

Value	Meaning
--------------	----------------

msNormal	
----------	--

msColumn	
----------	--

msRow	
-------	--

msAll	
-------	--

msURL	
-------	--

TNavigation

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Holds all settings of enhanced keyboard navigation in the grid

TNavigation.AdvanceAuto

[TNavigation](#)

property AdvanceAuto: Boolean **default** False; **Description**

When using masked input, automatically advances to the next cell when masked edit is completed.

TNavigation.AdvanceDirection

[TNavigation](#)

property AdvanceDirection: [TAdvanceDirection](#); **Description**

Sets the direction to be used with the AdvanceOnEnter navigation style.

TNavigation.AdvanceInsert

[TNavigation](#)

property AdvanceInsert: Boolean **default** False; **Description**

If true, allows insertion of rows when last cell is entered when in AdvanceOnEnter mode

TNavigation.AdvanceOnEnter

[TNavigation](#)

property AdvanceOnEnter: Boolean **default** False; **Description**
Next cell is selected after pressing enter while editing current cell.

TNavigation.AllowClipboardAlways

[TNavigation](#)

property AllowClipboardAlways: Boolean **default** False; **Description**

When true, automatic clipboard shortcut handling works irrespective of the grid editing options. When false, clipboard handling only works on editable cells.

TNavigation.AllowClipboardColGrow

[TNavigation](#)

property AllowClipboardColGrow: Boolean **default** False; **Description**

When true, the number of columns can automatically increase if more data is pasted in the grid than columns available.

TNavigation.AllowClipboardRowGrow

[TNavigation](#)

property AllowClipboardRowGrow: Boolean **default** False; **Description**

When true, the number of columns can automatically increase if more data is pasted in the grid than rows available.

TNavigation.AllowClipboardShortCuts

[TNavigation](#)

property AllowClipboardShortCuts: Boolean **default** False; **Description**

If true, enables Ctrl-Ins, Shift Ins, Ctrl-Del keys or also Ctrl-V, Ctrl-X, Ctrl-C to perform clipboard actions

TNavigation.AllowDeleteRow

[TNavigation](#)

property AllowDeleteRow: Boolean **default** False; **Description**

DEL key will delete row if true. OnAutoDeleteRow event is called on DEL

TNavigation.AllowFmtClipboard

[TNavigation](#)

property AllowFmtClipboard: Boolean **default** False; **Description**

When true, cells are copied to the clipboard in a custom clipboard format holding also the cell properties

TNavigation.AllowInsertRow

[TNavigation](#)

property AllowInsertRow: Boolean **default** False; **Description**

INS key will insert row if true. OnAutoInsertRow event is called on INS

TNavigation.AllowRTFClipboard

[TNavigation](#)

property AllowRTFClipboard: Boolean **default** False; **Description**
When true, rich text clipboard operation is used when available.

TNavigation.AllowSmartClipboard

[TNavigation](#)

property AllowSmartClipboard: Boolean **default** False; **Description**

When true allows automatic replication of single cell to multiple cells through clipboard copy & paste, as well as 2 cell difference based auto increment paste for selected cells. Difference based auto increment works for numbers as well as dates.

TNavigation.AlwaysEdit

[TNavigation](#)

property AlwaysEdit: Boolean **default** False; **Description**

Shows the inplace editor always (must to be used when additional inplace editors are used instead of goAlwaysShowEdit)

TNavigation.AutoComboDropSize

[TNavigation](#)

property AutoComboDropSize: Boolean **default** False; **Description**

When true, the inplace combobox editor dropdown list automatically sizes to show the largest string it holds

TNavigation.AutoGotoIncremental

[TNavigation](#)

property AutoGotoIncremental: Boolean **default** False; **Description**

If true, allows jumping to rows by keypress in sorted column in incremental lookup mode.

TNavigation.AutoGotoWhenSorted

[TNavigation](#)

property AutoGotoWhenSorted: Boolean **default** False; **Description**

When true, allows jumping to rows with single keypress in sorted column.

TNavigation.CopyHTMLTagsToClipboard

[TNavigation](#)

property CopyHTMLTagsToClipboard: Boolean **default** False; **Description**

When true, HTML tags are also copied to the clipboard in all clipboard operations.

TNavigation.CursorWalkEditor

[TNavigation](#)

property CursorWalkEditor: Boolean **default** False; **Description**

When true, allows changing of the current edit cell to the next of previous cell when edit position is either at the end of the edit text and right cursor key is pressed or if edit position is in the beginning of the edit text and left cursor key is pressed.

TNavigation.HomeEndKey

[TNavigation](#)

property HomeEndKey: [THomeEndAction](#); **Description**

Sets the action caused by pressing Home / End in the grid. This can be either moving to the first / last column or to the first / last row.

TNavigation.ImproveMaskSel

[TNavigation](#)

property ImproveMaskSel: Boolean **default** False; **Description**

Immediately puts the caret in the first position for a mask edit input in the grid.

TNavigation.InsertPosition

[TNavigation](#)

property InsertPosition: [TInsertPosition](#); **Description**

Can be either pInsertBefore or pInsertAfter. It determines if a row is inserted before or after the current row when pressing Ins with AllowAutoInsert enabled.

TNavigation.KeepHorizScroll

[TNavigation](#)

property KeepHorizScroll: Boolean **default** False; **Description**

When true, changing row selection when the grid is horizontally scrolled does not scroll back to leftmost column.

TNavigation.LineFeedOnEnter

[TNavigation](#)

property LineFeedOnEnter: Boolean **default** False; **Description**

When true, enter is used in the multiline inplace editor to do a linefeed instead of leaving the inplace editor. When false, Ctrl-Enter is need to do a linefeed in a multiline inplace editor and Enter leaves the inplace editor and optionally moves to the next cell when AdvanceOnEnter is true.

TNavigation.MoveRowOnSort

[TNavigation](#)

property MoveRowOnSort: Boolean **default** False; **Description**

When true, moves the grid display to the new position of the active cell after sort.

TNavigation Events

[TNavigation](#)

[Alphabetically](#)

[Legend](#)

TNavigation Events

[TNavigation](#)

[.By object](#)

[Legend](#)

TNavigation Methods

[TNavigation](#) [Alphabetically](#)

[Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TNavigation Methods

TNavigation

.By object

Legend

Assign

Destroy

GetNamePath

TNavigation Properties

[TNavigation](#)

[Alphabetically](#)

[Legend](#)

In TNavigation

- [AdvanceAuto](#)
- [AdvanceDirection](#)
- [AdvanceInsert](#)
- [AdvanceOnEnter](#)
- [AllowClipboardAlways](#)
- [AllowClipboardColGrow](#)
- [AllowClipboardRowGrow](#)
- [AllowClipboardShortCuts](#)
- [AllowDeleteRow](#)
- [AllowFmtClipboard](#)
- [AllowInsertRow](#)
- [AllowRTFClipboard](#)
- [AllowSmartClipboard](#)
- [AlwaysEdit](#)
- [AutoComboDropSize](#)
- [AutoGotoIncremental](#)
- [AutoGotoWhenSorted](#)
- [CopyHTMLTagsToClipboard](#)
- [CursorWalkEditor](#)
- [HomeEndKey](#)
- [ImproveMaskSel](#)
- [InsertPosition](#)
- [KeepHorizScroll](#)
- [LineFeedOnEnter](#)
- [MoveRowOnSort](#)

TNavigation Properties

TNavigation

By object

Legend

- [AdvanceAuto](#)
- [AdvanceDirection](#)
- [AdvanceInsert](#)
- [AdvanceOnEnter](#)
- [AllowClipboardAlways](#)
- [AllowClipboardColGrow](#)
- [AllowClipboardRowGrow](#)
- [AllowClipboardShortCuts](#)
- [AllowDeleteRow](#)
- [AllowFmtClipboard](#)
- [AllowInsertRow](#)
- [AllowRTFClipboard](#)
- [AllowSmartClipboard](#)
- [AlwaysEdit](#)
- [AutoComboDropSize](#)
- [AutoGotoIncremental](#)
- [AutoGotoWhenSorted](#)
- [CopyHTMLTagsToClipboard](#)
- [CursorWalkEditor](#)
- [HomeEndKey](#)
- [ImproveMaskSel](#)
- [InsertPosition](#)
- [KeepHorizScroll](#)
- [LineFeedOnEnter](#)
- [MoveRowOnSort](#)

TNodeClickEvent

Unit

advgrid

TNodeClickEvent=**procedure**(Sender: TObject; ARow, ARowreal: **Integer**) **of object**;

TNodeType

Unit

advgrid

TNodeType=(cnFlat, cn3D, cnGlyph, cnLeaf); **Description**

Value

Meaning

cnFlat	Flat node indicator
cn3D	3D node indicator
cnGlyph	Glyph node
cnLeaf	Leaf open / Leaf close sort indicator

TOleDragDropEvent

Unit

advgrid

TOleDragDropEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; data: **string**; var Allow: Boolean) **of object**;

TOleDragOverEvent

Unit

advgrid

TOleDragOverEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**; var Allow: Boolean) **of object**;

TOleDragStartEvent

Unit

advgrid

TOleDragStartEvent=**procedure**(Sender: TObject; ARow, ACol: **Integer**) **of object**;

TOleDragStopEvent

Unit

advgrid

TOleDragStopEvent=**procedure**(Sender: TObject; OLEffect: **Integer**) **of object**;

TOleDropColEvent

Unit

advgrid

TOleDropColEvent=**procedure**(Sender: TObject; ARow, ACol, DropCol: **Integer**) **of object**;

TOleDroppedEvent

Unit

advgrid

TOleDroppedEvent=**procedure**(Sender: TObject; ARect: TGridRect) **of object**;

TOnResizeEvent

Unit

advgrid

TOnResizeEvent=**procedure**(Sender: TObject) **of object**;

TPrintBorders

Unit

advgrid

TPrintBorders=(pbNoborder, pbSingle, pbDouble, pbVertical, pbHorizontal, pbAround, pbAroundVertical, pbAroundHorizontal, pbCustom); **Description**

Value	Meaning
pbNoborder	No borders printed
pbSingle	Single line border for cells
pbDouble	Double line border for cells
pbVertical	Vertical borders only
pbHorizontal	Horizontal borders only
pbAround	Only border around grid
pbAroundVertical	Only right & left vertical line
pbAroundHorizontal	Only horizontal top & bottom line
pbCustom	Custom border settings

TPrintMethod

Unit

advgrid

TPrintMethod=(prPreview, prPrint, prCalcPrint, prCalcPreview); **Description**

Value	Meaning
prPreview	Preview mode for printing
prPrint	Actual printing mode
prCalcPrint	Calculating printing mode to calculate nr. of pages
prCalcPreview	Calculating preview printing mode

TPrintPosition

Unit

advgrid

TPrintPosition=(ppNone, ppTopLeft, ppTopRight, ppTopCenter, ppBottomLeft, ppBottomRight, ppBottomCenter); **Description**

Value	Meaning
ppNone	The item is not printed
ppTopLeft	The item is printed at the top left side of the page
ppTopRight	The item is printed at the top right side of the page
ppTopCenter	The item is printed at the top center of the page
ppBottomLeft	The item is printed at the bottom left side of the page
ppBottomRight	The item is printed at the bottom right side of the page
ppBottomCenter	The item is printed at the top center of the page

TPrintSettings

[Hierarchy](#) [Properties](#)

[Methods](#)

Unit

advgrid

Description

Holds all printout settings

TPrintSettings.Borders

[TPrintSettings](#)

property Borders: [TPrintBorders](#); **Description**

The Borders property can be :pbNoborder : no border is printedpbSingle : single line width border is printedpbDouble : double line width border is printedpbVertical : only vertical single line borders are printedpbHorizontal : only horizontal single line borders are printedpbAround : only border around the grid is printedpbAroundVertical : only outer vertical borders of the grid are printedpbAroundHorizontal : only outer horizontal borders of the grid are printed

TPrintSettings.BorderStyle

[TPrintSettings](#)

property BorderStyle: [TPenStyle](#); **Description**
Line drawing style for border of type TPenStyle

TPrintSettings.Centered

[TPrintSettings](#)

property Centered: Boolean; **Description**

When true, printout is centered on the page.

TPrintSettings.ColumnSpacing

[TPrintSettings](#)

property ColumnSpacing: **Integer**; **Description**

Controls distance to skip between 2 columns in units of 1/10mm.

TPrintSettings.Date

[TPrintSettings](#)

property Date: [TPrintPosition](#); **Description**

The position has the type TPrintPosition with following values
ppNone : no date is printed
ppTopLeft : data is printed in top left corner
ppTopRight : date is printed in top right corner
ppTopCenter : date is printed centered on top
ppBottomLeft : date is printed in bottom left corner
ppBottomCenter : date is printed centered at bottom
ppBottomRight : date is printed in bottom right corner

TPrintSettings.DateFormat

[TPrintSettings](#)

property DateFormat: **string**; **Description**

Holds the date formatting parameter. Default value is dd/mm/yyyy.

TPrintSettings.FitToPage

[TPrintSettings](#)

property FitToPage: [TFitToPage](#); **Description**

- fpNever : never use page fitting.- fpAlways : always fit to page, no matter what the scalefactor is- fpGrow : only fit to page by increasing font size / column widths- fpShrink : only fit to page by decreasing font size / column widths- fpCustom : call the OnFitToPage event, to query for allowing pagefit with calculated scalefactor.

TPrintSettings.FixedHeight

[TPrintSettings](#)

property FixedHeight: **Integer**; **Description**

Height of rows for printout in 0.1mm. Overrides the auto calculated row height if UseFixedHeight is true

TPrintSettings.FixedWidth

[TPrintSettings](#)

property FixedWidth: **Integer**; **Description**

Width of columns for printout in 0.1mm. Overrides the auto calculated column width if UseFixedWidth is true

TPrintSettings.Font

[TPrintSettings](#)

property Font: [TFont](#); **Description**

Sets font of printout

TPrintSettings.FooterFont

[TPrintSettings](#)

property FooterFont: [IFont](#); **Description**

Sets font for the footer

TPrintSettings.FooterSize

[TPrintSettings](#)

property FooterSize: **Integer**; **Description**

Controls distance to skip at end of page in units of 1/10mm.

TPrintSettings.HeaderFont

[TPrintSettings](#)

property HeaderFont: [TFont](#); **Description**
Sets font for the header

TPrintSettings.HeaderSize

[TPrintSettings](#)

property HeaderSize: **Integer**; **Description**

Controls distance to skip at start of page in units of 1/10mm.

TPrintSettings.JobName

[TPrintSettings](#)

property JobName: **string**; **Description**

Sets title for print job in printer spooler

TPrintSettings.LeftSize

[TPrintSettings](#)

property LeftSize: **Integer**; **Description**

Controls distance to skip at left side of page in units of 1/10mm.

TPrintSettings.MaxPagesOffset

[TPrintSettings](#)

property MaxPagesOffset: **Integer**; **Description**

Sets the offset of the total nr. of pages printed.

TPrintSettings.NoAutoSize

[TPrintSettings](#)

property NoAutoSize: Boolean; **Description**

If true, disables the automatic column sizing to optimize paper use and retain full column text visibility but uses column sizes proportional to column sizes on display and prints using wordwrapping if wordwrap property is true

TPrintSettings.NoAutoSizeRow

[TPrintSettings](#)

property NoAutoSizeRow: Boolean; **Description**

When true, no automatic row height calculation is performed and the row height of the printed grid is proportional to the on screen row height.

TPrintSettings.Orientation

[TPrintSettings](#)

property Orientation: [TPrinterOrientation](#); **Description**

Sets printer orientation to landscape or portrait (defined in the Printers unit)

TPrintSettings.PageNr

[TPrintSettings](#)

property PageNr: [TPrintPosition](#); **Description**

The position has the type TPrintPosition with following valuesppNone : no page number is printedppTopLeft : page number is printed in top left cornerppTopRight : page number is printed in top right cornerppTopCenter : page number is printed centered on topppBottomLeft : page number is printed in bottom left cornerppBottomCenter : page number is printed centered at bottomppBottomRight : page number is printed in bottom right corner

TPrintSettings.PageNumberOffset

[TPrintSettings](#)

property PageNumberOffset: **Integer**; **Description**

Sets the offset of the page numbers printed.

TPrintSettings.PageNumSep

[TPrintSettings](#)

property PageNumSep: **string**; **Description**

Sets the separator between actual printed page and total nr. of pages. If PageNumSep is a zero length string, total number of pages is not printed. Default value is '/'

TPrintSettings.PagePrefix

[TPrintSettings](#)

property PagePrefix: **string stored** True; **Description**
Sets the prefix for page numbering for printout

TPrintSettings.PageSuffix

[TPrintSettings](#)

property PageSuffix: **string**; **Description**

Sets the suffix for page numbering for printout

TPrintSettings.PrintGraphics

[TPrintSettings](#)

property PrintGraphics: Boolean; **Description**
When true, graphics are printed

TPrintSettings.RepeatFixedCols

[TPrintSettings](#)

property RepeatFixedCols: Boolean; **Description**

When true, printout of fixed columns is repeated on each page.

TPrintSettings.RepeatFixedRows

[TPrintSettings](#)

property RepeatFixedRows: Boolean; **Description**

When true, printout of fixed rows is repeated on each page.

TPrintSettings.RightSize

[TPrintSettings](#)

property RightSize: **Integer**; **Description**

Controls distance to skip at right side of page in units of 1/10mm.

TPrintSettings.RowSpacing

[TPrintSettings](#)

property RowSpacing: **Integer**; **Description**

Controls distance to skip between 2 columns in units of 1/10mm.

TPrintSettings.Time

[TPrintSettings](#)

property Time: [TPrintPosition](#); **Description**

The position has the type TPrintPosition with following values

ppNone : no title is printed

ppTopLeft : title is printed in top left corner

ppTopRight : title is printed in top right corner

ppTopCenter : title is printed centered on top

ppBottomLeft : title is printed in bottom left corner

ppBottomCenter : title is printed centered at bottom

ppBottomRight : title is printed in bottom right corner

TPrintSettings.Title

[TPrintSettings](#)

property Title: [TPrintPosition](#); **Description**

The position has the type TPrintPosition with following values

- ppNone : no title is printed
- ppTopLeft : title is printed in top left corner
- ppTopRight : title is printed in top right corner
- ppTopCenter : title is printed centered on top
- ppBottomLeft : title is printed in bottom left corner
- ppBottomCenter : title is printed centered at bottom
- ppBottomRight : title is printed in bottom right corner

TPrintSettings.TitleLines

[TPrintSettings](#)

property TitleLines: [TStringList](#); **Description**

TitleLines is a stringlist that can be used instead of TitleText to hold a multiple title lines

TPrintSettings.TitleSpacing

[TPrintSettings](#)

property TitleSpacing: **Integer**; **Description**

Controls the space between the grid title and the actual grid data in 0.1mm units

TPrintSettings.TitleText

[TPrintSettings](#)

property TitleText: **string**; **Description**

TitleText holds a single line title only. When using multiline titles, use the TitleLines property

TPrintSettings.UseFixedHeight

[TPrintSettings](#)

property UseFixedHeight: Boolean; **Description**

If enabled, uses value FixedHeight, else auto calculation is done

TPrintSettings.UseFixedWidth

[TPrintSettings](#)

property UseFixedWidth: Boolean; **Description**

If enabled, uses value FixedWidth, else auto calculation is done

TPrintSettings Events

[TPrintSettings](#)

[Alphabetically](#)

[Legend](#)

TPrintSettings Events

[TPrintSettings](#)

[.By object](#)

[Legend](#)

TPrintSettings Methods

[TPrintSettings](#) [Alphabetically](#)

[Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TPrintSettings Methods

[TPrintSettings](#)

[.By object](#)

[Legend](#)

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TPrintSettings Properties

[TPrintSettings](#)

[Alphabetically](#)

[Legend](#)

In TPrintSettings

- [Borders](#)
- [BorderStyle](#)
- [Centered](#)
- [ColumnSpacing](#)
- [Date](#)
- [DateFormat](#)
- [FitToPage](#)
- [FixedHeight](#)
- [FixedWidth](#)
- [Font](#)
- [FooterFont](#)
- [FooterSize](#)
- [HeaderFont](#)
- [HeaderSize](#)
- [JobName](#)
- [LeftSize](#)
- [MaxPagesOffset](#)
- [NoAutoSize](#)
- [NoAutoSizeRow](#)
- [Orientation](#)
- [PageNr](#)
- [PageNumberOffset](#)
- [PageNumSep](#)
- [PagePrefix](#)
- [PageSuffix](#)
- [PrintGraphics](#)
- [RepeatFixedCols](#)
- [RepeatFixedRows](#)
- [RightSize](#)
- [RowSpacing](#)
- [Time](#)
- [Title](#)
- [TitleLines](#)
- [TitleSpacing](#)
- [TitleText](#)
- [UseFixedHeight](#)
- [UseFixedWidth](#)

TPrintSettings Properties

[TPrintSettings](#)

[By object](#)

[Legend](#)

- [Borders](#)
- [BorderStyle](#)
- [Centered](#)
- [ColumnSpacing](#)
- [Date](#)
- [DateFormat](#)
- [FitToPage](#)
- [FixedHeight](#)
- [FixedWidth](#)
- [Font](#)
- [FooterFont](#)
- [FooterSize](#)
- [HeaderFont](#)
- [HeaderSize](#)
- [JobName](#)
- [LeftSize](#)
- [MaxPagesOffset](#)
- [NoAutoSize](#)
- [NoAutoSizeRow](#)
- [Orientation](#)
- [PageNr](#)
- [PageNumberOffset](#)
- [PageNumSep](#)
- [PagePrefix](#)
- [PageSuffix](#)
- [PrintGraphics](#)
- [RepeatFixedCols](#)
- [RepeatFixedRows](#)
- [RightSize](#)
- [RowSpacing](#)
- [Time](#)
- [Title](#)
- [TitleLines](#)
- [TitleSpacing](#)
- [TitleText](#)
- [UseFixedHeight](#)
- [UseFixedWidth](#)

TRadioClickEvent

Unit

advgrid

TRadioClickEvent=**procedure**(Sender: TObject; ACol, ARow, Aldx: **Integer**) **of object**;

TRawCompareEvent

Unit

advgrid

TRawCompareEvent=**procedure**(Sender: TObject; Col, Row1, Row2: **Integer**; var Res: **Integer**) of **object**;

TRowChangingEvent

Unit

advgrid

TRowChangingEvent=**procedure**(Sender: TObject; OldRow, NewRow: **Integer**; var Allow: Boolean) **of object**;

TRowSizeEvent

Unit

advgrid

TRowSizeEvent=**procedure**(Sender: TObject; ARow: **Integer**; **var** Allow: Boolean) **of object**;

TScrollHintEvent

Unit

advgrid

TScrollHintEvent=**procedure**(Sender: TObject; ARow: **Integer**; var hintstr: **string**) **of object**;

TScrollHintType

Unit

advgrid

TScrollHintType=(shNone, shVertical, shHorizontal, shBoth); **Description**

Value	Meaning
shNone	No hints shown during scroll
shVertical	Hints shown during vertical scroll
shHorizontal	Hints shown during horizontal scroll
shBoth	Hints shown during horizontal & vertical scroll

TScrollType

Unit

advgrid

TScrollType=(ssNormal, ssFlat, ssEncarta); **Description**

Value	Meaning
ssNormal	Normal scrollbars
ssFlat	Flat scrollbars
ssEncarta	Encarta style scrollbars

TSizeWhileTyping

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Controls automatic inplace editor sizing capabilities

TSizeWhileTyping.Height

[TSizeWhileTyping](#)

property Height: Boolean **default** False; **Description**

Change height of cell when text becomes too high for cell while editing (wordwrap mode)

TSizeWhileTyping.Width

[TSizeWhileTyping](#)

property Width: Boolean **default** False; **Description**

Change width of cell when text becomes too wide for cell while editing

TSizeWhileTyping Events

[TSizeWhileTyping](#) [Alphabetically](#)

[Legend](#)

TSizeWhileTyping Events

[TSizeWhileTyping](#) [By object](#)

[Legend](#)

TSizeWhileTyping Methods

[TSizeWhileTyping](#) [Alphabetically](#) [Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TSizeWhileTyping Methods

[TSizeWhileTyping](#) [By object](#)

[Legend](#)

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TSizeWhileTyping Properties

[TSizeWhileTyping](#) [Alphabetically](#) [Legend](#)

In TSizeWhileTyping

- [Height](#)
- [Width](#)

TSizeWhileTyping Properties

[TSizeWhileTyping](#) [By object](#) [Legend](#)

- [Height](#)
- [Width](#)

TSortBlankPosition

Unit

advgrid

TSortBlankPosition=(blFirst, blLast); **Description**

Value

Meaning

blFirst Sets the position where blank cells should be during sort as first

blLast Sets the position where blank cells should be during sort as last

TSortDirection

Unit

advgrid

TSortDirection=(sdAscending, sdDescending); **Description**

Value	Meaning
--------------	----------------

sdAscending	Sorting is in ascending order
-------------	-------------------------------

sdDescending	Sorting is in descending order
--------------	--------------------------------

TSortSettings

[Hierarchy](#)

[Properties](#)

[Methods](#)

Unit

advgrid

Description

Sets various settings that affect sorting in the grid

TSortSettings.AutoColumnMerge

[TSortSettings](#)

property AutoColumnMerge: Boolean; **Description**

Automatically merge columns again after sorting based on the column indexes in the MergedColumns list

TSortSettings.AutoFormat

[TSortSettings](#)

property AutoFormat: Boolean; **Description**

When true, the grid tries to automatically guess the format of the data to be sorted.

TSortSettings.BlankPos

[TSortSettings](#)

property BlankPos: [TSortBlankPosition](#); **Description**

Sets the position where blank cells should be positioned during a sort.

TSortSettings.Column

[TSortSettings](#)

property Column: **Integer**; **Description**
Sets the main sort column index.

TSortSettings.Direction

[TSortSettings](#)

property Direction: [TSortDirection](#); **Description**
Sets the main sort direction

TSortSettings.DownGlyph

[TSortSettings](#)

property DownGlyph: [TBitmap](#); **Description**

Sets the glyph to indicated descending sort order on the column header

TSortSettings.FixedCols

[TSortSettings](#)

property FixedCols: Boolean; **Description**

When true, fixed columns are also sorted. When false, the fixed column contents are not affected by the sort.

TSortSettings.Full

[TSortSettings](#)

property Full: Boolean; **Description**

When true, columns from left to right are taken into account during the sort when cell values are identical starting from the main sort column.

TSortSettings.IgnoreBlanks

[TSortSettings](#)

property IgnoreBlanks: Boolean; **Description**

When true, blank cells are ignored during sorting and always put as last rows after the sort.

TSortSettings.IndexShow

[TSortSettings](#)

property IndexShow: Boolean; **Description**

When true, the grid shows indexed sort indicators in the column headers. Mouse left click sets the main sort index while mouse shift leftclicks add additional sort indexes.

TSortSettings.NormalCellsOnly

[TSortSettings](#)

property NormalCellsOnly: Boolean; **Description**

When SortNormalCellsOnly is true, fixed columns are not affected by the sort.

TSortSettings.Row

[TSortSettings](#)

property Row: **Integer**; **Description**

Sets the row index of the fixed row that should display the sort indicator and trigger the column sorting by a mouse click.

TSortSettings.Show

[TSortSettings](#)

property Show: Boolean; **Description**

If true, sorting when clicking on column headers is enabled and the sort indicators are displayed in the column headers.

The default sort indicator is a triangle, but custom bitmaps can be defined to indicate the sorting direction with the [UpGlyph](#) and [DownGlyph](#) properties

TSortSettings.SingleColumn

[TSortSettings](#)

property SingleColumn: Boolean; **Description**

When true, only cells within the column to be sorted are rearranged in sorted order leaving other columns unaffected.

TSortSettings.UpGlyph

[TSortSettings](#)

property UpGlyph: [TBitmap](#); **Description**

Sets the glyph to indicated ascending sort order on the column header

TSortSettings Events

[TSortSettings](#)

[Alphabetically](#)

[Legend](#)

TSortSettings Events

[TSortSettings](#)

[.By object](#)

[Legend](#)

TSortSettings Methods

[TSortSettings](#) [Alphabetically](#)

[Legend](#)

Derived from TPersistent

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TSortSettings Methods

[TSortSettings](#)

[.By object](#)

[Legend](#)

[Assign](#)

[Destroy](#)

[GetNamePath](#)

TSortSettings Properties

[TSortSettings](#)

[Alphabetically](#)

[Legend](#)

In TSortSettings

- [AutoColumnMerge](#)
- [AutoFormat](#)
- [BlankPos](#)
- [Column](#)
- [Direction](#)
- [DownGlyph](#)
- [FixedCols](#)
- [Full](#)
- [IgnoreBlanks](#)
- [IndexShow](#)
- [NormalCellsOnly](#)
- [Row](#)
- [Show](#)
- [SingleColumn](#)
- [UpGlyph](#)

TSortSettings Properties

TSortSettings

By object

Legend

- AutoColumnMerge
- AutoFormat
- BlankPos
- Column
- Direction
- DownGlyph
- FixedCols
- Full
- IgnoreBlanks
- IndexShow
- NormalCellsOnly
- Row
- Show
- SingleColumn
- UpGlyph

TSortStyle

Unit

advgrid

TSortStyle=(ssAutomatic, ssAlphabetic, ssNumeric, ssDate, ssAlphaNoCase, ssAlphaCase, ssShortDateEU, ssShortDateUS, ssCustom, ssFinancial, ssAnsiAlphaCase, ssAnsiAlphaNoCase, ssRaw, ssHTML, ssImages, ssCheckBox); **Description**

Value	Meaning
ssAutomatic	Sort style automatically guessed by grid
ssAlphabetic	Alphabetic sort
ssNumeric	Numeric sort
ssDate	Date sort
ssAlphaNoCase	Case insensitive alphabetic sort
ssAlphaCase	Case sensitive alphabetic sort
ssShortDateEU	Short European date sort
ssShortDateUS	Short US date sort
ssCustom	Custom sort
ssFinancial	Financial number sort
ssAnsiAlphaCase	Ansi case sensitive alphabetic sort
ssAnsiAlphaNoCase	Ansi case insensitive alphabetic sort
ssRaw	Raw sort
ssHTML	Sort ignoring HTML tags
ssImages	Image index based sort

TSpinClickEvent

Unit

advgrid

TSpinClickEvent=**procedure**(Sender: TObject; ACol, ARow, AValue: **Integer**; UpDown: Boolean) **of object**;

TStretchMode

Unit

advgrid

TStretchMode=(noStretch, Stretch, StretchWithAspectRatio, Shrink, ShrinkWithAspectRatio);

Description

Value

Meaning

noStretch No stretching is applied to pictures

Stretch Pictures are stretched to fit in the current cell width and height

StretchWithAspectRatio Pictures are stretched to fit in the current cell width and height, taking the aspect ratio into account

Shrink Only shrinking is used to fit an image in a cell. When the picture is smaller than the cell, no stretching occurs

ShrinkWithAspectRatio Only shrinking is used to fit an image in a cell. The aspect ratio is taken into account.

TUpdateColumnSizeEvent

Unit

advgrid

TUpdateColumnSizeEvent=**procedure**(Sender: TObject; ACol: **Integer**; var AWidth: **Integer**) **of object**;

TVAIignment

Unit

BaseGrid

TVAIignment=(vtaTop, vtaCenter, vtaBottom);

Description

Value	Meaning
vtaCenter	Vertical center alignment
vtaTop	Vertical top alignment
vtaBottom	Vertical bottom alignment

TWidthArray

Unit

advgrid

TWidthArray=**array**[0..MAXCOLUMNS]of **SmallInt**;

Scope

- Published
- Protected

Accessibility

- Read-only

Hierarchy

Exception

Hierarchy

[TASGComboBox](#)

Hierarchy

[TASGDropSource](#)

Hierarchy

[TASGDropTarget](#)

Hierarchy

[TAsgEditBtn](#)

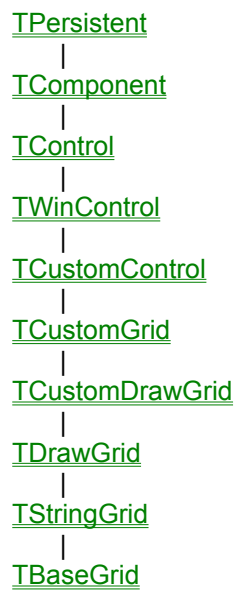
Hierarchy

[TAsgSpinEdit](#)

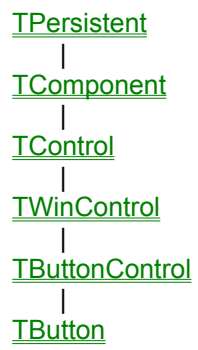
Hierarchy

[TAsgUnitEditBtn](#)

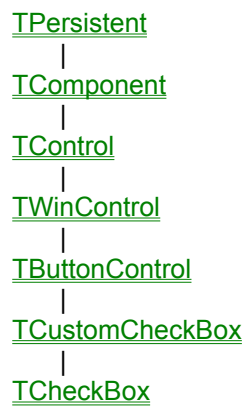
Hierarchy



Hierarchy



Hierarchy



Hierarchy

TPersistent

|

TCollection

Hierarchy

TPersistent

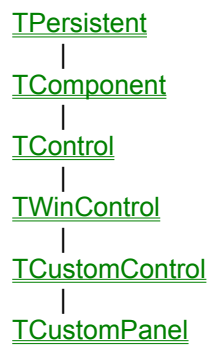
|

TCollectionItem

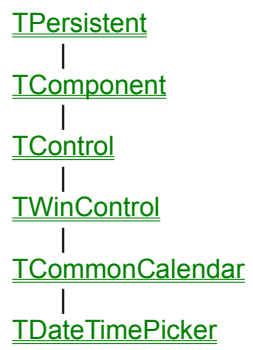
Hierarchy

TPersistent
|
TComponent

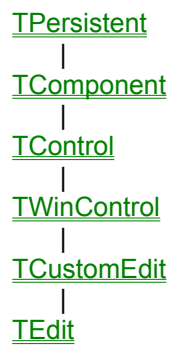
Hierarchy



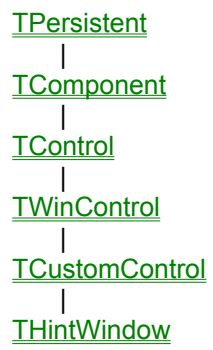
Hierarchy



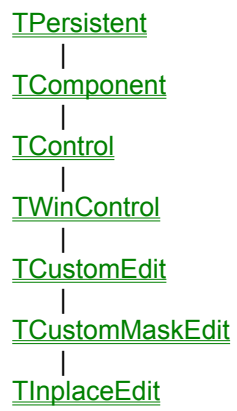
Hierarchy



Hierarchy



Hierarchy



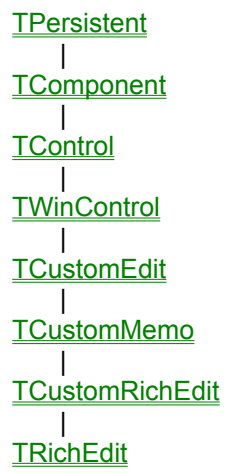
Hierarchy

TObject

Hierarchy

TPersistent

Hierarchy



Hierarchy

